Distributional Analysis of Entities

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik der
Universität Stuttgart zur Erlangung der Würde eines Doktors der
Philosophie (Dr. phil.) genehmigte Abhandlung

Vorgelegt von
Abhijeet Gupta
aus Udaipur, Indien

Hauptberichter:	Prof. Dr. Sebastian Padó
Mitbetreuer:	Prof. Dr. Gemma Boleda
Mitberichter:	Prof. Dr. Simone Ponzetto
Tag der mündlichen Prüfung:	12.05.2021

Institut für Maschinelle Sprachverarbeitung (IMS)
der
Universität Stuttgart

2022

"कोई तुम्हे तब तक नहीं हरा सकता, जब तक तुम खुद से ना हार जाओ "

–सुल्तान

(13.08.2019)

To my parents, my wife and Aryadit

Abstract

Arguably, one of the most important aspects of natural language processing is natural language understanding which relies heavily on lexical knowledge. In computational linguistics, modelling lexical knowledge through distributional semantics has gained considerable popularity. However, the modelling is largely restricted to generic lexical categories (typically common nouns, adjectives, etc.) which are associated with coarse-grained information i.e., the category *country* has *a boundary*, *rivers* and *gold deposits*. Comparatively, less attention has been paid towards modelling entities which, on the other hand, are associated with fine-grained real-world information, for instance: the entity *Germany* has precise properties such as, *(GDP - 3.6 trillion Euros), (GDP per capita - 44.5 thousand Euros)* and *(Continent - Europe)*.

The lack of focus on entities and the inherent latency of information in distributional representations warrants greater efforts towards modelling entity related phenomena and, increasing the understanding about the information encoded within distributional representations. This work makes two contributions in that direction:

(a) We introduce a semantic relation – *Instantiation*, a relation between entities and their categories, and distributionally model it to investigate the hypothesis that distributional distinctions *do* exist in modelling entities versus modelling categories within a semantic space. Our results show that in a semantic space: 1) entities and categories are quite distinct with respect to their distributional behaviour, geometry and linguistic properties; 2) Instantiation relation is recoverable by distributional models; and, 3) for lexical relational modelling purposes, categories are better represented by the centroids of their entities instead of their distributional representations constructed directly from corpora.

(b) We also investigate the potential and limitations of distributional semantics for the purpose of Knowledge Base Completion, starting with the hypothesis that fine-grained knowledge *is* encoded in distributional representations of entities during their meaning construction. We show that: 1) fine-grained information of entities is encoded in distributional representations and can be extracted by simple data-driven supervised models as attribute-value pairs; 2) the models can predict the entire range of fine-grained attributes, as seen in a knowledge base, in one go; and, 3) a crucial factor in determining success in extracting this type of information is *contextual support* i.e., the extent of contextual information captured by a distributional model during meaning construction.

Overall, this thesis takes a step towards increasing the understanding about entity meaning representations in a distributional setup, with respect to their modelling and the extent of knowledge inclusion during their meaning construction.

Zusammenfassung

Einer der möglicherweise wichtigsten Aspekte der Maschinellen Sprachverarbeitung ist das automatische Sprachverstehen, das stark auf lexikalisches Wissen angewiesen ist. In der Computerlinguistik hat die Modellierung lexikalischen Wissens durch distributionelle Semantik erhebliche Popularität erlangt. Allerdings ist die Modellierung weitgehend auf generische lexikalische Kategorien beschränkt (typischerweise Substantive, Adjektive etc.), die mit grobkörnigen Informationen assoziiert sind, d. h., die Kategorie *Land* hat *eine Grenze*, *Flüsse* und *Goldvorkommen*. Dagegen wurde vergleichsweise wenig Aufmerksamkeit der Modellierung von Entitäten gewidmet, die mit feinkörnigen Informationen über die reale Welt verknüpft sind, wie zum Beispiel: die Entität *Deutschland* hat präzise Eigenschaften wie z. B. *(BIP - 3,6 Billionen Euro), (BIP pro Kopf - 44,5 Tausend Euro) und (Kontinent - Europa)*.

Der fehlende Fokus auf Entitäten und die inhärente Latenz von Informationen in distributionellen Repräsentationen erfordert größere Anstrengungen zur Modellierung entitätsbezogener Phänomene und ein besseres Verständnis für die in distributionellen Repräsentationen kodierten Informationen. Die vorliegende Arbeit leistet zwei Beiträge in diese Richtung:

(a) Wir führen eine semantische Relation ein - *Instanziierung (Instantiation)*, eine Relation zwischen Entitäten und ihrem Kategorien, und modellieren sie distributionell, um die Hypothese, dass distributionelle Unterschiede zwischen der Modellierung von Entitäten und der Modellierung von Kategorien in einem semantischen Raum tatsächlich existieren, zu untersuchen. Unsere Ergebnisse zeigen, dass in einem semantischen Vektorraum: 1) Entitäten und Kategorien sich in Bezug auf ihre Distribution, ihre Geometrie und ihre linguistischen Eigenschaften deutlich unterscheiden; 2) die Relation der Instanziierung durch distributionelle Modelle extrahierbar ist;

und 3) für Zwecke der lexikalischen relationalen Modellierung Kategorien durch die Zentroide ihrer Entitäten besser repräsentiert werden als durch ihre distributionellen Repräsentationen, die direkt aus Korpora konstruiert werden.

(b) Wir untersuchen ebenso das Potenzial und die Grenzen der distributionellen Semantik zum Zwecke der Vervollständigung von Wissensdatenbanken, ausgehend von der Hypothese, dass feinkörniges Wissen in distributionellen Repräsentationen von Entitäten während ihrer Bedeutungskonstruktion kodiert wird. Wir zeigen, dass: 1) feinkörnige Informationen über Entitäten in distributionellen Repräsentationen kodiert sind und durch einfache datengetriebene überwachte Modelle als Attribut-Wert-Paare extrahiert werden können; 2) die Modelle die gesamte Bandbreite an feinkörnigen Attributen, wie sie in einer Wissensdatenbank zu finden sind, in einem Durchgang vorhersagen können; und 3) ein entscheidender Faktor zur Bestimmung des Erfolgs bei der Extraktion dieser Art von Informationen die kontextuelle Unterstützung ist, d.h., die Menge der kontextuellen Informationen, die von einem distributionellen Modell während der Bedeutungskonstruktion erfasst werden.

In ihrer Gesamtheit ist die vorliegende Dissertation ein Schritt zur Erweiterung des Verständnisses von Bedeutungsrepräsentationen von Entitäten in einem distributionellen Aufbau, bezogen auf ihre Modellierung und die Menge an Wissen, das während ihrer Bedeutungskonstruktion einbezogen wird.

Acknowledgements

First and foremost, I would like to express my sincere thanks to my *Doktorvater*, Prof. Dr. Sebastian Padó. My doctoral journey had a steep learning curve, and he was with me every step of the way. He has been my guiding beacon every time I needed to find the right direction. He gave me the freedom to explore, learn from my mistakes and pulled me out of quite a few tight spots. His perfect grasp of concepts, from the most basic to the most complex, in a multitude of disciplines has taught me how to objectively look at research problems, connect the (oftentimes fuzzy) dots and critically examine the outcomes. His high intellect is equally matched by his kindness, generosity, patience and his ability to empathise with (my) problems at all levels – as a student, a researcher and personal. He has been an outstanding mentor, both professionally and personally, and for this I will remain forever grateful.

I would also like to thank my second advisor, my *Doktormutter*, Prof. Dr. Gemma Boleda. She has played a fundamental role in motivating and conceptually building my doctoral research work from the ground up. Her acute observations, not just while designing the modelling approaches but also at evaluating quantitative and qualitative analyses, gave me valuable insights in understanding the big picture as well as working out the minute details. Throughout my work, she too has pro-actively encouraged me to explore, learn, stay focussed and stood by me during tougher times. For all of this, I feel deeply indebted.

I would like to thank Prof. Marco Baroni, with whom I had the chance to collaborate on my first major publication.

I would next like to thank my parents, Dr. Rakesh Gupta and Dr. Shubha Gupta, who always seem to have an unlimited supply of hope, optimism and positivity. In every endeavour I have ever undertaken, they have always

walked the extra mile beside me and sometimes carried me through as well. As I grow older, I am increasingly humbled by their unconditional love and unshakable faith in me.

Another heartfelt thanks, to my wife Meenal Gupta, who has not just patiently and consistently encouraged me to reach my goals but has also courageously and silently shared my burdens wherever possible. Her character, habits and values inspire me to be a better version of myself.

I am very grateful for the blessings of my 91 year old grandmother (Aaji), Manik Sabnis, who has been more like a mother to me. She now often fails to remember my name but has never failed to ask if my PhD is over and whether I am eating properly.

My sincerest thanks to my friend Devendra Natani. He has been more of a brother to me, than a friend, for the last 16 years. There are some bonds that words cannot express but, my deepest gratitude towards him in being instrumental in my journey to find my way back home.

I would not be at this juncture without the help, support, advise and friendship of several colleagues and people I know. I thank you all from the bottom of my heart. Dipit Nanawati, my cousin – for making sure I remain consistently motivated and focussed on writing my thesis. Christian Scheible – for helping me settle down professionally and personally – in a new department and a new country, for encouragingly guiding me through my first set of experiments and for explaining every (seemingly) complex concept in the easiest way possible. A hearty shout-out to Max Kisselew and Tanmoy Mukherjee for being great friends and my partners in the joys and miseries of the doctoral life. Anuj Katiyal and Abinash Mohapatra – my friends in need, always ready to show me the silver lining. Katalin Ötvös – for her vivaciousness as well as kindness and generosity, specially at unexpected moments. Evgeny Kim, Maja Buljan, Jason Utt and Kyle Richardson – for the many interesting conversations we had about work and life in general. Diego Frassinelli and Gabriella Lapesa – for always meeting me with a smile, kind words and sound advise. Last but not the least, Sebastian Wohlrapp – for being one of the most helpful, trustworthy

and optimistic persons I have come across in daily life.

A special thanks to Anubhav Kaushik, Bharti Gaur, Charchit Bapna, Preksha Kothari Bapna – for being as warm, understanding, supportive and accommodating as only family could be.

Finally, in the spirit of supervised learning, I would also like to thank all those confounders who have unwittingly contributed to my learning and growth, both professionally and personally.

Contents

List of Figures

List of Tables

List of Abbreviations

CCKB	**C**ollaboratively **C**onstructed **K**nowledge **B**ase
CL	**C**omputational **L**inguistics
DH	**D**istributional **H**ypothesis
DS	**D**istributional **S**emantics
DSMs	**D**istributional **S**emantic **M**odels
KB	**K**nowledge **B**ase
KBC	**K**nowledge **B**ase **C**ompletion
KR	**K**nowledge **R**epresentation
LS	**L**exical **S**emantics
NE	**N**amed **E**ntity
NEC	**N**amed **E**ntity **C**lassification
NER	**N**amed **E**ntity **R**ecognition
NERC	**N**amed **E**ntity **R**ecognition and **C**lassification
NLP	**N**atural **L**anguage **P**rocessing
NLU	**N**atural **L**anguage **U**nderstanding
ML	**M**achine **L**earning
SW	**S**emantic **W**eb

List of Publications

- A. Gupta, G. Boleda, M. Baroni, S. Padó. Distributional vectors encode referential attributes. *In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015, Lisbon, Portugal*.

- A. Gupta, G. Boleda, M. Baroni, S. Padó. Mapping conceptual features to referential properties. *3rd International ESSENCE Workshop: Algorithms for Processing Meaning, 2015, Barcelona, Spain*.

- G. Boleda, A. Gupta, S. Padó. Instances and Concepts in Distributional Space. *In Proceedings of 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL) - p111, 2017, Valencia, Spain*.

- A. Gupta, G. Boleda, S. Padó. Distributed Prediction of Relations for Entities: The Easy, The Difficult, and The Impossible. *In Proceedings of 6th Joint Conference on Lexical and Computational Semantics (*SEM), 2017, Vancouver, Canada*.

- A. Gupta, G. Boleda, S. Padó. Instantiation. *Arxiv Pre-print, arXiv.org, 2018.*.

Author Contribution: In the above publications, I, Abhijeet Gupta, have contributed towards: 1) defining the research problems; 2) designing and implementing the approaches related to: data collection, modelling and analyses; and finally, 3) writing the publications.

Chapter 1

Introduction

Lexical Semantics (LS) is the study of the meaning of individual words and the lexical relations they occur in, within the vocabulary of a language (Cruse et al., 1986). The words, that LS is primarily concerned with in a linguistic expression, contain *open-set elements* i.e. lexical roots or meaning bearing elements which can easily be substituted with other words[1]. The lexical relations are critical for semantic inferencing i.e. from a linguistic point-of-view, they constitute the core of semantic data modelling. Due to this, LS is an important component of natural language understanding which in turn supplements Natural Language Processing (NLP). As a consequence, the Computational Linguistics (CL) community has extensively invested in computationally modelling lexical relations where distributional semantics has gained immense popularity as a modelling framework.

1.1 Distributional Semantics

The Distributional Hypothesis (DH), proposed by Harris (1954) and Firth (1957), states that words with similar meaning appear in similar contexts. Distributional Semantics (DS), built on the foundations of DH, establishes this hypothesis by using Distributional Semantic Models (DSMs) to construct word meaning representations that are harvested from large corpora by aggregating over the distribution of words with other words in context

[1]For example, in the sentence *Germany developed cars* – (*Germany* | *develop* | *car*) are open-set elements with other substitutions like (*France, Heinz* | *make, sell* | *wine, ketchup*). A linguistic expression can also contain closet-set elements like, affixes and grammatical markers such as articles, conjunctions, prepositions, etc. which usually have few or no possibilities of substitution.

(Turney and Pantel, 2010). The distributional meaning representations are known to capture many aspects of word meaning. These meaning representations, being numeric in nature, are easily represented in a linear algebraic framework within an n-dimensional space, also called as a *semantic (or distributional) space* (Schütze, 1993), where each word meaning representation is a *vector*. The vectors allow for easy quantification of the semantics of a word and of the relations between words (Erk, 2012). Due to quick construction, easy quantification of semantics and general applicability, DSMs can be used by different types of language modelling tasks.

1.1.1 Distributional Modelling of Lexical Relations

Lexical relations, in language, are semantic relations between concepts expressed through words. For computational modelling of language, identifying the characteristics of semantic relations by evaluating the similarities (and dissimilarities) between words (and their underlying concepts) has been considered an important area of research (Chaffin and Herrmann, 1984). Therefore, the CL community has exhaustively modelled classical lexical relations like *synonymy, antonymy, polysemy, hypernymy and hyponymy* (amongst others). The words in these relations are typically adjectives and nouns but can be other generic word categories as well. In a seminal work, Hearst (1992) proposes the concept of using lexico-syntactic patterns from text to acquire lexical *is_a* relations automatically. And, since these lexical relations are known to occur with a high degree of co-occurring contexts (Turney, 2006), DS has come out as a go-to framework lexical relation modelling. We list out a few lexical relations that are popularly distributionally modelled:

1. **Synonymy**: implies that two words map to the same meaning and substituting one with the other does not change the meaning of the expressions under consideration (Cruse et al., 1986), for example (*quick – fast – speedy*). Note that synonymy mostly implies near-synonymy, i.e. the words differ from each other in finer aspects of

their denotation, as absolute synonymy is rare and limited to dialectal variations and technical terms (Clark, 1992) like, (*underwear* (US) – *pants* (UK)) or (*groundhog – woodchuck*) respectively. Synonymous words are primarily in a *paradigmatic* relation because they usually do not co-occur but, are observed within the same surrounding contexts like, *the delivery service is [quick|fast]* (Sahlgren, 2006).

2. **Antonymy**: is the opposite of synonymy, indicating a semantic relation where two words have opposite (or contrasting) meanings (Deese, 1964), for example (*hot – cold*) or (*fire – ice*). In recent times, the definition of antonymy has been further extended into two broad categorizations: gradable and non-gradable antonymy (Jones, 2003). The first example is of the gradable type where *more* hot means *less* cold; gradable implies a continuum and negation between the words; ultimately there being a mid-point where the meaning of both converges. The second example is of the non-gradable type, reflecting a categorical and opposing sense between two words i.e., more fire does *not* mean less ice, however, they might be used linguistically to convey the opposing extremes of a concept like 'temperature' or 'anger'. According to Sahlgren (2006), antonymous words can be in a paradigmatic relation – *hot coffee* and *cold coffee*, as well as a syntagmatic relation because they are known to frequently co-occur in text (Justeson and Katz, 1992), therefore, having similar contexts – *He was prescribed a **hot** and **cold** compress* and *Her temperament was a mix of **fire** and **ice**.*

   Both synonymy and antonymy are symmetric relations i.e., in the relation *R* between two words (or expressions) *a* and *b*, if *R(a,b)* then *R(b,a)*.

3. **Hypernymy**: on the other hand, is an example of an asymmetric relation between two words in a taxonomical hierarchy. It is a semantic relation between a hypernym (a superordinate) and its hyponym (subordinate) where the semantic scope of the hypernym is more

abstract as compared to its hyponym. For example, in the word-pair (animal-dog) – *animal* is a hypernym of *dog* and, the hyponym necessarily implies the meaning of its hypernym i.e., *dog is an animal* and its true for all dogs. The asymmetry comes in the form of the reverse not being true i.e. all animals are not necessarily dogs (they can be cats, rats, etc.). So, a hyponym can have only one hypernym but, a hypernym can have multiple hyponyms (also called co-hyponyms). The words in a hypernymy relation can be both in a syntagmatic or paradigmatic relation. The reason behind the importance given to hypernymy modelling is that it is considered to be a key organizational component of semantic memory (Murphy, 2004) and consequently very useful for tasks and applications that require structured lexical resource development and inferential reasoning (Angeli et al., 2016; Garg et al., 2019).

Such lexical relations (as above) can be modelled by constructing DSMs that can learn to identify the degree of semantic similarity (or relatedness)[2] between their word-pairs. The accuracy of models can be then evaluated against human judgements. While the relations can be modelled independently (Landauer and Dumais, 1997; Turney, 2006), they can also be modelled jointly to distinguish contrastive relations like, (synonymy – antonymy) or (hypernymy – hyponymy) (Shwartz and Dagan, 2016; Arora et al., 2020).

The most difficult aspect about relational modelling is for the distributional models to learn the distinction between similarity and relatedness (by association or opposition) [3]. To deal with it, efforts have been chiefly made in two directions:

[2] See Section 2.1.3 to understand the distinction between the two.

[3] The distributional representation for *dark* gives a high similarity score with representations for both *white* and *black*. This does not mean that the meaning of *dark* is the same as *white*, in fact, they are contradictory concepts. However, due to occurring in similar contexts, the representation of *dark* might have a higher similarity score with *white* as compared to *black* – which is more similar in meaning.

- towards creating datasets that linguistically capture the finer aspects that are a reflection of the increase in understanding of the distributional behaviour of the relations. While most datasets are parallel to well recognized linguistic resources (like, WordNet (Miller, 1995)), they include carefully crafted negative relations or syntactic cues to induce accurate learning (Roller and Erk, 2016; Santus et al., 2016). Additional variations are seen in the form of inclusion of (concept-activity) lexical relations (Baroni and Lenci, 2011), a larger variety of relations (Santus et al., 2015) or features related to each datapoint (Seitner et al., 2016).

- towards creating optimized word-embeddings that can be used to increase the accuracy of the models, for example, Yih et al. (2012) push a word and its antonyms away from each other in a semantic space, Lazaridou, Baroni, et al. (2015) on the other hand bring synonyms closer (as compared to antonymys) in the semantic space by using Wordnet, Nguyen et al. (2017) create distributional representations that can capture (hypernym–hyponym) distributional hierarchy and so on.

Understanding the linguistic phenomena through the distributional modelling of lexical relations also in-part serves the modelling needs of diverse tasks like Selectional Preferences (Erk et al., 2010) to get information about argument fillers, word sense disambiguation (WSD) (McCarthy and Carroll, 2003), lexical substitution (McCarthy and Navigli, 2009), semantic role labelling (Roth and Woodsend, 2014), predicting human plausibility judgements (Resnik, 1996), textual entailment (Baroni et al., 2012; Sadrzadeh et al., 2018). Which, in turn are used for NLP applications like, commonsense based semantic inferencing through linguistic expressions (Narisawa et al., 2013), model verification and evaluation in question-answering (Khot et al., 2018), information extraction (Angeli et al., 2015), machine translation (Rocktäschel et al., 2015) and, NLP oriented research streams, for

example, bio-medical text processing (Moen and Ananiadou, 2013) and psycholinguistics (Lenci, 2008).

1.1.2 Challenges of Distributional Relational Modelling

What is evident is that DS has been used to address a plethora of linguistic phenomena and tasks in CL. And, it has come to be accepted as an elegant and successful framework (Baroni et al., 2014a). However, the simplicity of the distributional framework has also given rise to its critique in the CL community:

- In LS, distributional modelling has so far been predominantly restricted only to lexical relations between categories. A *category* is a concept or, a class of objects grouped together based on common characteristics, thus, typically represented by common nouns, verbs, adjectives like *man, grass, country, run, bake, build, yellow, good, etc* (Miller and Hristea, 2006) (a more detailed definition can be found in Section 1.2).

  One reason for this has to do with the fact that text usually has an abundance of patterns reflecting lexical generalizations. For statistical approaches, where large amounts of data is a prerequisite, studying linguistic phenomena related to categories makes an ideal test-bed (Hearst, 1992; Morris and Hirst, 2004). The other reason lies in the meaning construction methodology of DSMs. Since the distributional representations are an aggregation of distributions of contexts: 1) the constructed meaning is only an approximation of the actual meaning (Dinu and Baroni, 2014); and, 2) since DSMs capture both syntagmatic and paradigmatic relations, the semantic neighbours identified by DSMs suggest that they provide coarse-grained representation of lexical meaning (Lenci, 2018) and they lack the *preciseness* in features deemed important for semantic grounding (Murphy, 2004).

Due to this, the notion of meaning in DS framework becomes similarity based which is not considered to be a sufficient substitute for identity based notions (Fodor and Lepore, 1999). Hence, lexical relations involving words other than categories (like *entities*, discussed next in Section 1.2) remain under-addressed.

- Another challenge is the interpretability of distributional meaning representations. The DSMs construct meaning representations in the form of vectors, which are nothing but an array of numbers. The fact that these representations perform well on various lexical modelling tasks provides sufficient evidence that they successfully capture the semantic (or syntactic) associations that a word has with other words; but this information resides in a latent manner within the vector. It is difficult to interpret this information other than by the way of task-specific evaluations (Murphy et al., 2012a) or by estimating the common lexical (thus, semantic) features between two representations that are near to each other in the semantic space (Boleda and Erk, 2015).

  Ways to improve interpretability involve creating sparse meaning representations which can be distributional (Murphy et al., 2012a) or non-distributional (Faruqui and Dyer, 2015). The former having the disadvantage of not being entirely (only relatively more) interpretable and the latter requires tedious feature engineering from additional lexical resources which makes it a comparatively complicated and specialized task as compared to the DS approach.

Thus, as of now, it remains difficult to assess what aspects of word meaning are systematically captured (or not) by distributional meaning representations.

1.2 Entities

Formal semantics prescribes to lexical concepts as *classes*, which refer to sets of objects bound together by a finite schema and largely represented by common nouns (Montague, 1970). And, in CL, they are popularly called **categories** (Hopper and Thompson, 1984). Considering that a category is a set of objects, a specific object from that set is called an *instance* of that category. An instance has distinct characteristics and properties in the real-world which distinguishes it from all the other objects contained within its category (or even other similar categories). In CL and its sub-fields these instances, often referred through *proper nouns*, are called **entities** [4]. The difference between the two is explained through the following example:

1. Category: the word *dog*, a common noun, is a lexical concept signifying an animal who barks, has four legs and a tail. These properties are common to *any* dog within the observable universe.

2. Entity: *Lassie* – the dog, a proper noun and an instance of the lexical concept *dogs*, is a fictional character who is a Rough Collie, with brown and white fur along with all the other properties of a *dog* i.e. Lassie also barks, has four legs and a tail.

Entities in text are mostly observed in a descriptive sense through mentions about their properties, actions or activities which concern them and possibly other entities as well. This makes entities a rich source of information, specially in terms of world knowledge (Limaye et al., 2010). The information is typically seen in the form of specific attributes and values which help in grounding the underlying concepts to the real-world, in contrast to their corresponding category. For example, consider the two expressions: *a country has a GDP* [5] and *Germany has a GDP of 3*

[4] Entities (or instances), when specifically referred to by proper nouns, are also called *Named Entities* – a term coined by the Information Extraction community, as explained in Section 2.5

[5] GDP stands for Gross Domestic Product, a financial marker used for assessing the economic growth of a country.

trillion euros. The former associates an attribute with a category as one of its properties and gives a generalized information that *all* countries have a GDP. The latter, on the other hand, associates an entity *Germany*, i.e. an instance of the *country* category, with an attribute-value pair that grounds it to the real world with information of a very precise nature. Due to the referential aspects of these attribute-value pairs we call them *referential attributes* or *fine-grained attributes*. On the other hand, attributes of a category provide generic information, hence, called *coarse-grained attributes*.

Entities are considered important for many NLP applications, whether they rely on domain knowledge or world knowledge, specially by the Knowledge Representation (KR) and Semantic Web (SW) communities. The KR community primarily works towards semantic grounding of text, i.e. linking text (specially entities) to the real world. Thus, it focuses specifically on modelling real-world knowledge aspects of meaning through entities and the relations between them.

One of the primary tasks for entity based applications is to first recognize them in text and subsequently type them to their classes – known as Named Entity Recognition and Classification (NERC). Traditional approaches, which are pattern based and often use hand-crafted rules and supplemental lexicons, are limited to performing NERC through coarse-grained contexts around entity mentions and string matching (Mikheev et al., 1999; Cucchiarelli et al., 1998). The fine-grained contexts, being precise in nature, are usually ignored by these approaches due to their inability to encode this information into a manageable set of patterns, rules and lists. One can see efforts to use fine-grained information, but these are restricted to using a pre-defined set of ontological classes for entity typing (Sekine and Nobata, 2004; Nadeau and Sekine, 2007). However, with the progress in development of sophisticated statistical approaches for meaning construction and language modelling, fine-grained information found in text is being increasingly utilized by state-of-the-art context driven distributional models for NERC (Shimaoka et al., 2017).

In addition to fine-grained NERC, the KR and SW communities have used distributionally constructed entity representations for tasks like entity relation prediction (Socher et al., 2013a), entity linking and slot filling (Blanco et al., 2015; Chen et al., 2015), semantic similarity identification (Šarić et al., 2012; Plank and Moschitti, 2013), semantic ontology identification (Fu et al., 2014), knowledge completion (Neelakantan et al., 2015) and common-sense reasoning (Jebbara et al., 2019).

1.2.1 Challenges Related to Entities

Given the wide-spread application of entities in NLP, they have not received their fair share of attention vis-a-vis analyses and evaluation of their semantic properties within a distributional setup. We specifically observe two entity related under-addressed phenomena in distributional lexical semantics:

1. Lexical relations of categories have been investigated extensively in CL. Contrastively, distributional modelling of the lexical relations that entities have with their immediate categories, for example: *Abraham Lincoln is a president* and *Germany is a country*, remain under-addressed.

   One plausible reason is the assumption that the analyses and conclusions for phenomena related to lexical generalizations would also stand true for entity oriented phenomena. However, this is not true because the lexical relation between a *category–category* indicates a concept-to-concept subsumption. Whereas, the lexical relation between an *entity–category* indicates the realization of a concept in the real world (usually) through a proper noun (Miller and Hristea, 2006). This distinction has also long been recoginized by the Formal Semantics and Ontology communities in theory (Schaerf, 1994; Gangemi et al., 2001) and its importance in connecting a real-world referent to a concept has been acknowledged as well within the KR community (MacGregor and Brill, 1992).

2. The success of DSMs in the area of knowledge prediction suggests the existence of fine-grained information in distributional representations of entities. In view of the Distributional Hypothesis this is surprising because DSMs use contextual information to build meaning representations and any information (including fine-grained) would typically get spread over large spans of text (across documents and time) and therefore, the knowledge extracted from such sources would likely take the form of generic knowledge (Baroni et al., 2014a). Therefore, while graded aspects of meaning are easily represented in a distributional space (Erk, 2010), for example – *Italy is more similar to France than Germany*, on the other hand, fine-grained information is considered to be difficult to express through distributional geometry and linear algebra (Bruni et al., 2012; Boleda and Herbelot, 2016), like *Germany has a population of 80 million* or *Germany is located in Europe*.

   Overall, the current state of DS, as it stands today, reflects that DS can address both linguistic generalities (of categories) as well as particularities (of entities) in terms of modelling. But, the encoding of fine-grained knowledge within distributional representations exists merely as an accepted notion i.e. seen in practice but *without* any further empirical clarifications on the nature or type of fine-grained information captured by distributional representations.

1.3 Research Questions and Contributions

The aim of this thesis is to investigate the properties of fine-grained information through distributional modelling of entities in a linguistic and an application oriented setup. The larger unexplored question that this thesis addresses is that *how* similar (or dissimilar) are entities and categories with respect to their distributional behaviour and geometry. To this purpose, we pose two research questions:

1. Can we model Instantiation distributionally?

   Instantiation is the relation between an instantiated entity and its category, typically occurring as a copular lexical relation in text and linguistic resources.

   Hypothesis: Instantiation can be modelled distributionally and Instantiation is functionally different from hypernymy.

   To test our hypothesis, we create a sizeable dataset containing (entity–category) word-pairs, for example, *Einstein – physicist*. We also create a *comparative* category oriented dataset on the lines of the classical lexical relation – *hypernymy*, containing (category- category) word-pairs, for example, *physicist – scientist*. We then plan to compare the distributional behaviour and geometry of Instantiation vs. hypernymy. The distributional behaviour is directly observable in phenomenon like difficulty of learning the lexical relation, the accuracy of classifying entities and categories of the same class, etc. The distributional geometry can be observed in evaluation of sub-spaces occupied by entities against that of categories.

   Note that Instantiation is only partially comparable to hypernymy in the sense that in both there is a subordinate-superordinate taxonomic relation between the linguistic units. Contrastively, the key difference is that the subordinate class in this relation is an entity and not a category, like in hypernymy. Due to the fine-grained information that entities occur with in text, Instantiation should be functionally different from hypernymy when modelled distributionally. Additionally, while such lexical patterns are addressed in knowledge prediction tasks by the KR community, their aim is to find entity or relation fillers to predict structured information (Freitas et al., 2014); which is fundamentally different from relational modelling.

   Contributions:

- To the best of our knowledge, we are the first to model the Instantiation relation distributionally.
- We show that Instantiation proper is recoverable by distributional models. And, that entities and categories are quite distinct with respect to their distributional behaviour, geometry and linguistic properties.
- We show that for Instantiation modelling, category representations derived from entity representations (centroids computed by averaging the entities instantiated by a category) are a better fit as compared to representations constructed for categories directly from the corpora.
- We also compare Instantiation vs hypernymy detection (by creating a semantically comparable hypernymy dataset) and provide empirical evidence which shows that the two relations, while appearing to be similar taxonomically, are linguistically and computationally distinct.
- We publicly release a dataset for Instantiation which is analogous to the state-of-the-art datasets released for other comparable relations like hypernymy and synonymy (Baroni and Lenci, 2011; Roller et al., 2014).

  The dataset is designed for handling the *lexical memorization* issues commonly faced during modelling of lexical relations (Levy et al., 2015b).

2. Is fine-grained information encoded in distributional representations?

   Hypothesis: Fine-grained information can be extracted from distributional representations of entities.

   To test this hypothesis, we extract fine-grained information from standard off-the-shelf distributed representations through supervised learning algorithms and represent them in the structured language of knowledge bases i.e. as attribute-value pairs; which can be *numeric*

or *categorical*. An overwhelming amount of work concentrates on *categorical* attributes. We, on the other hand, choose to focus on the prediction of *numerical* attributes of entities (Davidov and Rappoport, 2010)

The objective is to test whether the distributional entity representations capture the necessarily required semantic information to make such precise predictions confidently and in a generalizable manner. Additionally, we also focus on the following: 1) understanding the extent of granularity of information that can be captured by distributed representations i.e. *how* fine-grained can the predicted attributes be; 2) the type of information captured i.e. is the information domain centric with respect to geography, economy, etc.; and, 3) the aspects which determine the easiness and difficulty of predicting such information using machine learning techniques.

Contributions:

- We provide empirical evidence of fine-grained information being encoded within distributional representations. And, that it can be extracted in a structured attribute-value format by using supervised algorithms to a reasonable degree of accuracy.
- Unlike other studies who focus on a limited set of attributes (Buitelaar and Cimiano, 2008; Socher et al., 2013a), we induce full attribute-based description of entities.
- We explore two approaches for optimal prediction of fine-grained attributes. While other approaches compute the total performance across all attributes and rarely provide in-depth analyses of individual attributes, we, in contrast, follow a recently emerged direction of research that specifically aims at a better understanding of the nature and structure of information that is encoded within such representations (Lin et al., 2015; Herbelot and Vecchi, 2015; Xie et al., 2016; Beltagy et al., 2016). And, we present an assessment of the factors, beyond

data sparsity, that inhibit accurate prediction of certain classes of attributes.

- Lastly, we release state-of-the-art datasets as a substantial resource of fine-grained attribute information for entities for future work.

The entities in our datasets belong to well known personalities in the public sphere because we can easily build corpus representations for them since they are realized as proper nouns. Additionally, the structured data associated with them, that is available in the public domain (like in, WordNet and Freebase) provides an easily accessible reliable gold standard.

1.4 Thesis Structure

In Chapter 2, we elaborate upon the theoretical and applied foundations on which our work as well as the previous work (with which we compare our work) is built upon. We start by discussing distributional semantics, explaining how meaning is represented through distributional semantic models, how the meaning representations are evaluated and discuss the models through which our semantic space is built from. We touch upon the concepts of Clustering and its importance in exploratory data analysis, specially in our distributional setups in Chapter 3 and 4. We then discuss knowledge-bases, the prominence of entities in them and their problem of incompleteness – which motivates the work in Chapter 5. Finally we discuss Named Entity Recognition and Classification as a research area due to its comparability with our work in Chapter 3.

Chapter 3 is dedicated to the modelling the lexical relation of *Instantiation* in a distributional setup, which as a task has not been undertaken uptil now to the best of our knowledge; at least not in terms of an empirical exploration of the distributional behaviour of the relation as well as its elements (entities and categories). We discuss its motivation from previous work in formal as well as computational semantics and we contrast the task

of instantiation detection from other similar tasks in NLP. We then talk about our dataset design and follow it up with a distributional exploration of the elements involved in modelling the relation. Finally, we model instantiation detection and discuss our results and observations.

Chapter 4 is an extension of the previous chapter and built around questions that arose as a result of the instantiation modelling experiment. In particular, we address two points through mutually exclusive experiments. In the first experiment, we explore optimizing instantiation modelling by constructing an alternate representation for categories. In the second experiment, we compare instantiation with hypernymy detection to identify the differences in their distributional behaviour and show that the two relations are computationally distinct. The distinctions primarily arising due to entity representations being informationally different from categories.

Chapter 5 explores *what* makes entity representations informationally different than categories. To this purpose, we carry out an experiment to identify and extract the encoded fine-grained (referential) attributes from distributional representations of entities. We present empirical evidence on the numeric and categorical attribute classes that can be predicted successfully and a qualitative analysis of the properties due to which attribute prediction benefits (or, suffers) for certain attribute classes.

Chapter 6 presents an alternate approach to learning fine-grained categorical attributes. We present a model architecture that can not only predict those attributes which are not seen during training but is also more generic i.e., it can process diverse domains of varying sizes effectively. We conclude this chapter by listing the factors which are necessary for successful modelling of categorical attributes.

Chapter 7 presents the conclusions drawn from this research work and the possible future work that is required to further answer some open ended questions which are presently outside the scope of this work.

Chapter 2

Background

2.1 Distributional Semantics

One aim of Computational Linguistics (**CL**), as a research discipline, is to study and understand the semantics of language(s) to enhance language technologies and human-machine interactions. CL does this primarily through two streams: lexical semantics and compositional semantics. The former deals with addressing questions and linguistic phenomena related to the meaning of words and the latter explores the contribution of the meaning of individual words in complex phrases and sentences.

While the study of meaning of words and phrases has led to the development of many semantic theories over time, one theory in particular has become predominant in both CL and NLP communities in the recent past – **Distributional Semantics (DS)**. This is because DS provides the ability to easily quantify word-meaning in purely numerical terms. This numerical meaning representation can then be effectively used by statistical models for empirical validation of linguistic theories and their further applications in CL, as well as in NLP.

2.1.1 Representation of 'meaning' in Distributional Semantics

DS has its foundations in the **Distributional Hypothesis (DH)** (Harris, 1954). According to DH, words with similar meaning appear in similar contexts; therefore, the meaning of a word can be approximated by its distributional profile in text i.e. its contextual co-occurrence patterns. These

	CAT	DOG	LION
runs	2	5	2
hunts	3	1	8

TABLE 2.1: Hypothetical example of distributional information collected from a corpus

contexts are typically content words like: nouns, adjectives, verbs or any other word category, representing *concepts* integral to the semantics of the phenomenon being observed. The contexts together, in essence, represent the meaning of that word through its distributional history which can then be used as its semantic representation (Schütze and Pedersen, 1997).

These distributional semantic representations, being numeric, are easily expressed in a linear algebraic framework, where each word is represented as a **vector**. As a hypothetical example, let us assume that our vocabulary consists of only three nouns – *cat*, *dog* and *lion*, which are observed in a corpus with the following contexts: *runs* and hunts. Table 2.1, depicts these nouns and their contexts along with their distributions in the corpus. The distributions are collected by traversing the corpus and calculating the co-occurrence counts of contexts with the nouns. Through these counts we can represent the distributional vectors of the nouns: *cat*, *dog* and *lion* with column 1, 2 and 3 respectively.

There are certain conclusions that we can deduce right away from the distributions collected; that *cat*, *dog* and *lion* are similar in the sense that all of them run (as seen in 1st row). But, they are dissimilar from the point of view of hunting i.e. both *cat* and *lion* hunt whereas the *dog*, acknowledged as a domesticated animal, largely does not (as seen in 2nd row). Looking at the overall distributions, within the scope of the current contexts, *cat* and *lion* are **more** similar as compared to *dog*. This distributional information allows for a precise quantification of the meaning similarities between the three animals, represented geometrically as shown in Figure 2.1.

The figure shows a two-dimensional construction of the *cat*, *dog* and *lion* distributional vectors on a Cartesian plane with x and y co-ordinates

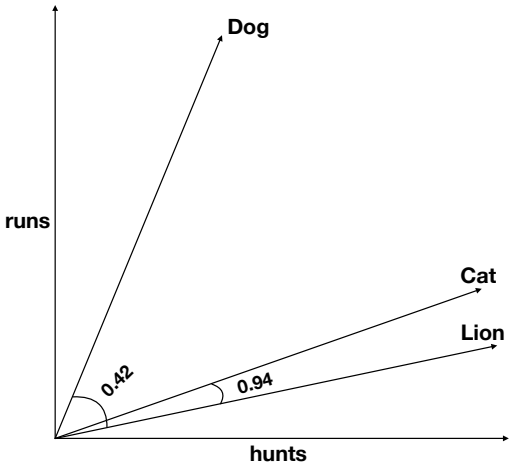


FIGURE 2.1: Geometric representation of the distributional vectors in Table 2.1

representing two contexts, *hunts* and *runs* respectively. The semantic similarity between the vectors can be measured by the distance between the vectors, by computing the cosine of the angle between them.

Cosine is a similarity metric which is mathematically a function of the the width (i.e., size) of the angle between the vectors, calculated by computing the normalized scalar product of the vectors. If the vectors contain positive values (i.e. co-occurrence counts / distributions), the cosine has a value-range between [0,1]. On the other hand, if the counts are transformed into other scores (see Section 2.1.2.1), the cosine value-range can be between [-1,1] as well. In either case, 0 and -1 indicate minimal semantic similarity with the former implying orthogonal vectors and the latter implying parallel vectors pointing in opposite directions. 1 indicates maximal semantic similarity, represented by parallel vectors pointing in the same direction.

From the given example, vectors for *dog* and *lion* have a cosine of 0.42, indicating low similarity through a wide angle. Whereas *cat* and *lion* have a cosine of 0.94, indicating high similarity with a narrow angle.

The vector is made up of parts called *components* where each component corresponds to an individual context. The total number of components is referred to as the *size* of that vector. In terms of linear algebra, the vector exists in an N-dimensional vector space where each dimension corresponds to a component and thus, to a context as well. In this way, the space can include an arbitrary number of vectors where one specific dimension of each vector maps to a specific context. For example, the first dimension of the vectors of *cat*, *dog* and *lion* is a common context: *runs*.

These vectors, within the vector space, are the *distributional meaning representations* of the words they represent. The vector space is also referred popularly as *semantic space* in CL.

2.1.2 Distributional models of meaning construction

The semantic space and its vectors are constructed by Distributional Semantic Models (DSMs). While DSMs come in many flavours, these can be divided into two broad categories:

2.1.2.1 Count-based DSMs

The *count* based models initialize vectors for each word by collecting its co-occurrence contextual counts, similar to the example discussed in Table 2.1. The choice of contexts relevant to the meaning of the word, in terms of type and number, are predetermined through a *window* within the vicinity of the word.

The contextual window is an important parameter of DSMs and it can precede, succeed or surround the word. The earliest instances of DSMs are referred to as *bag of words* [1] models because the contexts are collected from context window without considering *any* relational dispositions [2] of the

[1] The term *bag* here corresponds with its mathematical usage to refer to a *multiset* – a construct similar to a *set* but one that allows duplicates. For example, (dog, dog, cat, rain, rain) is a bag. The order of words is inconsequential and the bag: (rain, dog, cat, dog, rain) is equivalent to the former.

[2] Traditional *windows* are “distance" based, however as discussed towards the end of this section, modern variants also consider other relations to collect contexts.

word or word-order constraints (Schütze and Pedersen, 1997). Depending on the task at hand, these contexts can either be all the words within the window or belong to combinations of specific functional word categories, like *nouns*, *verbs*, *adjectives*, etc. The length of the window is variable and is generally seen as an experimental parameter; popularly referred as *window-size*. However, where the earlier approaches leaned towards larger window-sizes of around 20-100 words (Yarowsky, 1992; Niwa and Nitta, 1994; Schütze and Pedersen, 1997), the more recent works have started restricting themselves to contexts in the immediate vicinity of around 2-10 words. One reason, undoubtedly, is the availability of much larger corpora, leading to a comparatively larger coverage with smaller windows. The other reason is the steadfast empirical evidence in published state-of-the-art models which favours smaller windows for capturing both topical and relational information about a word (Sahlgren, 2008; Lenci, 2018).

Another important parameter in DSMs is the *dimensionality* of the semantic space. In general practice, the number of contexts collected from a corpus can be quite large. This results in the construction of a large semantic space i.e. the space has a very high number of dimensions usually ranging in thousands: Schütze (1993) use a 5000 dimensional space and, Lund and Burgess (1996b) report the informational analyses of vectors in spaces upto 70,000 dimensions. The count-based high-dimensional semantic spaces, thus constructed, have been found to suffer from the following: 1) such spaces are known to be problematic with respect to modelling because in such cases the semantic similarity between words is also affected by their relative frequency in the corpus, which according to *Zipf's Law* [3] is highly skewed (Lowe, 2001); and, 2) these spaces are also highly sparse because a large amount of words in the vocabulary do not co-occur as contexts, thereby, having a co-occurrence frequency of 0 (Turney and Pantel, 2010).

[3] The law states that the frequency of a word is (approximately) proportional to the reciprocal of its rank in a frequency list (Zipf, 2016). For example, in English, the ten most frequent words *the, be, of, and, to, a, in, have, that, it* constitute to slightly over one quarter of all tokens in the 100M British National Corpus.

To deal with the former, the raw frequencies are usually *transformed* into significance weighted scores by using methods like, Pointwise Mutual Information (PMI) [4] (Church and Hanks, 1990) and its other variations: like, Positive PMI [5] (Manning et al., 1999), Normalized PMI [6] (Bouma, 2009), various odd ratios and entropy based normalization (Landauer and Dumais, 1997). These scoring methods ensure that infrequent contexts are given more weightage as compared frequent contexts. Overall, weighted space transformation is known to produce a more effective representation of the meaning of the word as compared to simple co-occurrence counts (Kiela and Clark, 2014).

To mitigate the problems arising due to data sparseness and the *curse of dimensionality* – as dimensionality grows, the number of operations required to address the problem and the memory required to execute the operations grow exponentially (Oseledets and Tyrtyshnikov, 2009), the high dimensional semantic space can also be *reduced* to a lower (and more manageable) dimensional space through linear or non-linear algebraic methods, like Singular Value Decomposition (SVD) (Van Der Maaten et al., 2009, and references therein). This reduced dimensionality should ideally correspond to the intrinsic dimensionality of the original data i.e. the minimum number of parameters required to account for the observed properties of the data should remain intact within the reduced dimensions (Fukunaga, 2013). Note that Landauer and Dumais (1997) also claim that dimensionality reduction improves the semantic representation of high-dimensional spaces. The reduced space is called a *dense semantic space* and has been experimented with varying dimensionalities of 50, 100, 300, 500, 700, 900 and 1000 (Lapesa and Evert, 2014).

[4] **PMI** is the measure of association between two outcomes (x,y) belonging to two discrete random variables (X,Y). PMI can have a positive or negative value; a positive PMI indicates high co-occurrence of the variables, whereas, 0 indicates their independence and negative values indicates less- than-expected co-occurrence.

[5] **PPMI** – PMI with negative values rounded up to 0.

[6] **NPMI** – PMI normalized between [-1,1], where -1 stands for no co-occurrence, 0 for independence and 1 for complete co-occurrence.

Harris (1954), in addition to the DH, also postulates that the differences in meanings of words are a function of the differences in their distributional contexts. Therefore, once the vectors have been constructed, transformed and reduced, the degree of semantic similarity (or in general, relatedness) and dissimilarity between two words can be estimated by applying similarity based metrics (like cosine similarity) over their vectors (Lee, 1999). Note that there is no contradiction between similarity and dissimilarity here because both are inversely proportional in relation i.e. the *more* similar two words are, the *less* dissimilar they become (and vice-versa).

Types of count based DSMs. There can be many other variants of DSMs, in addition to the vanilla bag-of-words models. The variations are largely based on two distinctions (and their combinations): a) type of contexts; and, b) the methodology of constructing the semantic space from which the distributional representations are extracted.

Some of the other prominent models are: 1) document-based DSMs, like Latent Semantic Analysis (LSA) and Hyperspace Analogue of Language (HAL) (Landauer and Dumais, 1997; Lund and Burgess, 1996a), where word similarity is based on their occurrence in the same documents and paragraphs; 2) syntax-based DSMs, where the contexts are selected based on the dependencies of the word (Padó and Lapata, 2007; Baroni and Lenci, 2010); and 3) matrix based topical and word models, where the distributional data is first collected in the form of co-occurrence matrices and then the matrix is transformed to a dense matrix to produce informationally compact and richer distributional representations (Steyvers and Griffiths, 2007; Dinu et al., 2013).

The list above is not exhaustive and a reasonably detailed description and performance assessment of count-based family of DSMs can be found in the works of Kiela and Clark (2014), Lapesa and Evert (2014), and Levy and Goldberg (2014a). Evidently, what comes to light is that different DSMs excel at capturing different aspects of words-level semantic associations. And, one conclusion common to all such evaluations is that there is

no conclusive evidence of one DSM outrightly outperforming the rest in every scenario.

2.1.2.2 Predictive DSMs

Predictive DSMs estimate the *probability* of words occurring in the vicinity of other words, in text. The term *predictive* is derived from the ability of the model to predict a target word from its contexts (or the contexts from a target word). Unlike the count-based models that observe context frequencies to create meaning representations, the predictive models *optimize* the meaning representations of words based on their co-occurrence patterns in the corpus. Therefore, given enough data and a good objective function, a predictive model can give highly accurate word-meaning estimations. Once again, based on the DH that similar words occur in similar contexts, the predictive models learn to *assign* similar vectors to similar words.

Vectors and Embeddings. These vectors are also referred to as *embeddings* because their construction methodology originates from the Machine Learning stream where the vectors exist in a dense low-dimensional space (low – as compared to the typical space dimentionalities observed in CL). And, each vector component has features *embedded* into it during its construction. In the spirit of consistency, the recent trend has been to often use the term *embeddings*, interchangeably with *vectors*.

The predictive models are a relatively recent phenomena. They initially gained prominence as neural language models where the word embeddings were seen as by-products to the modelling objective of predicting: 1) the next word in a sequence (Bengio et al., 2003; Blitzer et al., 2005; Mnih and Hinton, 2009); 2) an unseen word external to an existing finite semantic space (Larochelle, Bengio, et al., 2006); and even, 3) predicting syntactic and semantic labels (Collobert and Weston, 2008).

Due to the high computational complexity (and inefficiency) of these models, their application to linguistics remained rather sparse and restricted to experimental implementations. However, their popularity has seen

an exponential resurgence with the models introduced by Mikolov et al. (2013a) and Pennington et al. (2014), which are computationally feasible, robust and have been proven to capture distributional features to the tune of the other state-of-the-art count-based models of the time.

The *Word2Vec* models by Mikolov et al. (2013a) are the pivotal models in the increased application of predictive DSMs. As shown in Figure 2.2, the two models (called CBOW and Skip-gram) are shallow neural networks i.e. without any hidden layers. The models accept word representations as input; initially constructed by generating a one-hot encoding (Murphy, 2012) for each word over the entire vocabulary. The dimensionality of the inputs and outputs for both models is equal. The models are essentially log-linear classifiers with a continuous projection layer and have an objective (function) of maximizing the conditional log-likelihood of the output. Because the models are computationally efficient, as compared to their parent neural models (Mikolov, 2007; Mikolov et al., 2009) and counterparts (Morin and Bengio, 2005; Bengio, LeCun, et al., 2007; Mnih and Hinton, 2009), they can be used to compute very high (and low) dimensional word representations over much larger data [7]. The resultant embeddings are known to work well at capturing both syntactic as well as semantic regularities (Mikolov et al., 2013b).

- **Continuous Bag-of-Words Model (CBOW)**

  In CBOW, the preceding and succeeding contexts words $w(t \pm i)$, within a symmetric window with the target word $w(t)$ in the middle, are summed (averaged, as a variant) into a common projection position. This projection is then used by the classifier in a *forward* pass to predict the target word. Thereafter, through *back-propagation* the model parameters and the input embeddings are adjusted to better predict the target word. The size of the symmetric window (i) is understood to be a hyper-parameter and has been experimented

[7] Mikolov et al. (2013a) reported using the Google news corpus of about 6 billion tokens

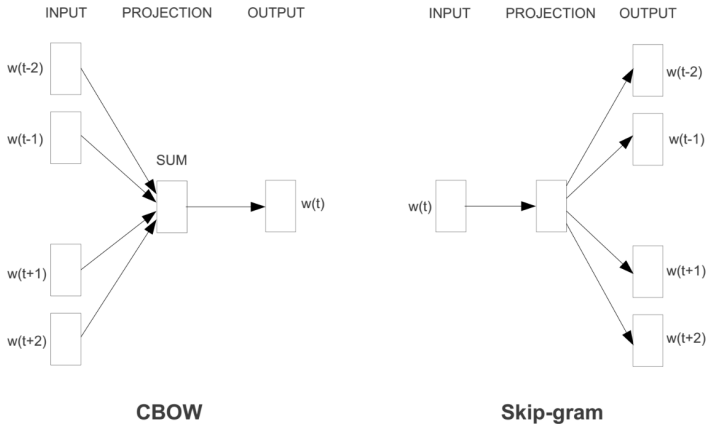


FIGURE 2.2: Word2Vec model architectures, as expressed in (Mikolov et al., 2013a). The target word is $w(t)$ and its surrounding contexts are $w(t \pm i)$. The CBOW model predicts a target word from its surrounding contexts whereas, the Skip-gram model predicts the surrounding contexts from the target words.

with varying sizes ranging from **[**2,10**]** contexts, finally tuned to an optimum size of 4.

C, in CBOW, implies that the contexts map to a *continuous* distributed representation and, *BOW* implies that the order of contexts, given as input, does not influence their projection.

- **Continuous Skip-gram Model**

  The Skip-gram model, on the other hand, takes a target word $w(t)$ as input, projects it and uses the projection to predict continuous context representations $w(t \pm i)$ within a surrounding symmetric window (i). Essentially, the conditional probabilities $p(w(t \pm i)|w(t))$ are computed from the corpus text to optimally set the parameters θ in $p(c|w;\theta)$; where (c) is the set of contexts for the word w.

  While the architecture is similar to CBOW, Skip-gram differs in one critical aspect: it predicts one context at a time, updates the target

word and then goes on to predict the next context i.e. each context-target pair is treated as one training sample. CBOW, contrastively, updates all the inputs at one go, therefore, making the entire set of (contexts + target) within the window one training sample. As an example, consider the following phrase: *The red roses*, where **red** is the target word and the rest are contexts. In Skip-gram, this constitutes to 2 training samples – $[(the, red), (roses, red)]$. Whereas, in CBOW its one train sample – $[(the + roses, red)]$.

The Skip-gram model prioritizes closer contexts through assigning higher weights to its parameters, since distant contexts contribute comparatively lesser to the meaning of the word. The size of the window is again a tuned hyper-parameter optimally set to a maximum of 5 preceding and succeeding contexts. Because, while an increase in size leads to observably better word representations, it also increases the computational complexity of the model.

At a broad level, Skip-gram is overall slower than CBOW in performance due to a higher computational complexity per (contexts + target). But, it does a better job of predicting infrequent words because for each word it is designed to predict the contexts, therefore, all target words tend to get equal attention. CBOW on the other hand is designed to predict the most probable word from contexts, therefore, frequent words get rich embeddings and the embeddings of rare words get smoothed-out over time (with the assumption that both frequent and rare words would have shared contexts) (Mikolov et al., 2013a).

2.1.2.3 Count-based models or Predictive models?

Baroni et al. (2014b) present a thorough systematic comparison between the two types of models on various lexical semantics tasks where the predictive models either outperform or are atleast equivalent in performance to the otherstate-of-the-art count-based models. Their recommendation to use

predictive models stems from the need to tune relatively less number of hyper-parameters for optimal model performance.

The predictive models, despite their simple architecture, also give state-of-the art results in tasks with complex phrase construction or those requiring semantic inferencing (Mikolov et al., 2013b; Le and Mikolov, 2014). These models have also been quite effective in addressing various multi-modal tasks as demonstrated by Socher et al. (2013b) and Frome et al. (2013).

However, one big advantage for count-based models is a larger degree of *transparency*. Each step of model construction, as described in Section 2.1.2.1, from distribution collection to space transformation(s) and reduction can be modularized and fine-tuned in a controlled and incremental manner. This is quite contrary to the predictive models which *implicitly* learn the parameters, thus, making them less transparent and their choices of tuning hyper-parameters prone to slight randomness. Moreover, Levy et al. (2015a) go on to argue that the Skip-gram model implicitly computes Pointwise Mutual Information (PMI) (Church and Hanks, 1990) and that much of the success of predictive models is due to the hyper-parameter settings which can also be accounted for in count-based models.

Despite the arguments in favour (or equivalency) of count-based models, predictive models have been the clear winners when it comes to a choice between the two due to their high-quality distributional representations, generic applicability, ease of construction and tunability of model parameters – both in terms of time and computational complexity.

2.1.3 Evaluation of Meaning Representations

By the way of their construction, vectors within a semantic space share contextual information. Therefore, whether a DSM is count-based or predictive, its evaluation is typically done by computing the semantic relatedness or semantic similarity between meaning representations of

lexical units or linguistic expressions in various lexical semantics challenges (Lee, 1999; Budanitsky and Hirst, 2006; Mohammad and Hirst, 2012).

Semantic relatedness between two words is characterized by the existence of *any* semantic relation between them. For example, *house* and *brick* or *car* and *wheel* (Budanitsky and Hirst, 2001). In other words, the two words have a correspondence in the relations they share with other words.

Semantic similarity, on the other hand, is a specific type of relatedness in the sense that it is a measure of *taxonomical similarity* between two words (Resnik, 1996) i.e. there is a correlation between the semantic properties of the words. For example, *oil* and *petrol* are similar because they are both sources of fuel and energy or *apples* and *oranges* both are types of fruits.

To empirically assess the model performance, the basic methodology is to compute semantic relatedness or similarity between meaning representations: 1) Intrinsically i.e. against a binary judgement or human annotated score(s); or, 2) Extrinsically, by applying distributional representations in various linguistic (or more generally, NLP) downstream tasks such as, Named entity recognition, Part-of-speech tagging, Syntactic chunking (Ghannay et al., 2016; Chiu and Nichols, 2016), Sentiment analysis (Schnabel et al., 2015) and, Semantic role labelling (Collobert et al., 2011).

Thereafter, an aggregated performance score can be generated through metrics based on Accuracy, F-score (Sokolova et al., 2006), Residuality or Correlation (Rosset et al., 2005; Asuero et al., 2006). These metrics are equally suitable for evaluating models mentioned in the next sections as well.

2.1.4 Role of Corpus

The choice of corpus plays a critical role in the effectiveness of a DSM. Traditionally, domain specific or task specific corpora were highly popular (Grefenstette, 1994; Hamon and Nazarenko, 2001). However, recent times have brought a clear shift from using specialized corpora to using large

scale open-domain corpora (Baroni and Lenci, 2010). The main motivation has been to increase contextual coverage with the view that: 1) understanding and analysis of word meaning can come only with the knowledge of all the contexts in which a word can occur in (Martinet, 1989); and, 2) for successfully modelling a lexical phenomenon through distributional methods, the inclusion of relevant context is necessary (Shwartz et al., 2017). A larger coverage ensures that the semantic behaviour of a word is more realistic, rather than specifically tailored according to the problem at hand. This also reduces data sparseness which has been known to adversely affect the performance of distributional models.

2.2 Classification

Classification is a technique that allows the determination (or, forecasting) of group membership of data instances (Kesavaraj and Sukumaran, 2013). In other words, its an approach that allows partitioning of data into categorical classes based on a certain set of conditions. Traditionally, these conditions could be specified heuristically, however, in the recent times data-driven approaches have gained considerable traction as well (Gan et al., 2018).

2.2.1 Supervised Classification

Definition: The aim of supervised classification is to train a classifier from labelled data D, containing a predefined number of classes c_n and datapoints x_n, where each datapoint x_i typically represents one class c_i (Santafe et al., 2015):

$$D = (c_1, x_1), ..(c_i, x_i), ..., (c_n, x_n) \tag{2.1}$$

Each datapoint x_i is characterized by n predictive variables:

$$x_i = (X_1, X_2, ..., X_n) \tag{2.2}$$

The learned classifier can be used to predict a class between $(c_1, ..., c_n)$ for a new data instance for which the class is unknown. The learning over the predictive variables should be generalizable and ideally, this datapoint should not be a part of the data used for training the classifier. The performance of the classifier can be judged through a scoring mechanism (or, evaluation metric) that can quantify the behaviour of the classifier over a task. Note that there is no specific scoring approach that universally suits all tasks. *Supervised* approaches are known to outperform the other alternatives, primarily because supervised models are trained over labelled data that (in theory) covers the scope of the problem being addressed in its entirety. The two other approaches (broadly) are: 1) *semi-supervised* classification, where both labelled and unlabelled data are utilized for classification. A classifier is initially trained on a relatively small amount of labelled data and then it is used to classify the unlabelled data, which can in turn be used iteratively as labelled data for improved learning (Gan et al., 2013); 2) *unsupervised* classification, that entails the partitioning of unlabelled data into meaningful groupings. In the age of internet, unlabelled data is easily available in large quantities, and therefore, this approach is comparatively easy to configure and cost effective as well (Krishnapuram et al., 2004). The trade-off, to the aforementioned advantages, comes in the form of accuracy and specificity i.e., the results from the approach are not always reliable and thus, are best suited for making generalizations about the structure of the data (Hebboul et al., 2015). Which is why the approach is usually used for the purposes of data exploration

2.2.2 Types of Supervised Classification

Classification is a predictive modelling problem i.e., the learning approach has to predict the outcome, given a certain set of variables as input. There are broadly three types of classifications:

Binary Classification. In this type of classification, the task is to classify between one class or the other i.e., there are only two classes and

every input datapoint can belong exclusively to only one class (Menon and Williamson, 2018). In other words, the model predicts the success of a binary Bernoulli trial (Weisstein, 2002), by predicting either a 0 or 1. A classic example of this classification is categorization of emails as *spam – 0* or *not spam – 1*.

Multi-Class Classification. In multi-class classification, the task is to assign a datapoint into one of the n observed classes. For instance, classifying a plant into one of the plant species. A simple approach to solving this task is to view this problem as a binary classification problem where a classifier can learn to distinguish between one class vs the rest of the classes, or alternately, compute pairwise scores for all classes and then selecting the class with the most winning two-class comparisons (Wu et al., 2003).

Multi-Label Classification. In the multi-label classification we have n observed classes as well, however, in this approach each datapoint can belong to more than one class. For example, object classification in an image – a bicycle, a person and an animal may occur together a single image which have to be identified. Here too, the binary approach of one vs. the rest can be easily applied or alternately, we can use a classifier that prescribes a probability score to each class, together totalling up 1 (Ghamrawi and McCallum, 2005)

2.2.3 Supervised Classifier: Logistic Regression

In this section, we describe the supervised classification technique through a simple logistic regression classifier. Let us consider a binary classification problem with two classes $y \epsilon \{c_1, c_2\}$ and the measured input data that is in the form of a real-valued n- dimensional vector $x = [d_1, d_2, ..., d_n]$. The task is to assign this vector to one of the two classes. This classifier is based on a logistic function (also called the *sigmoid* function) that can take any real-valued input and map it to a value between 0 and 1:

$$y = \frac{1}{1 + e^{-\epsilon}} \tag{2.3}$$

where, ϵ represents the function that contains the real- valued data used for model learning, which has to be mapped between 0 and 1. The objective of a logistic classifier is to predict the probability of the default class and consequently, the logistic regression classifier can be viewed as the conditional probability of predicting a class:

$$P(y|X) = \frac{1}{1 + e^{(\mathbf{W}^T X + b)}} \tag{2.4}$$

where, **W** is a real-valued weight matrix representing the d variable parameters, X represents the input instances $(x_1, x_2, ..., x_n)$ and b is the intercept vector. The dimensionality of the data instances (x_i) determines whether the problem is linear or non-linear. For instance, if the dimensionality is 1, then the number of variable parameters equals to 1 and the problem has a linear solution i.e., the classes can be partitioned by a straight line, so to speak. However, as the dimensionality increases, the number of variable parameters increase and the classes cannot be typically separated by one straight line.

Initially, at classifier training, **W** is typically initialised by zeros or randomly (using small fractional values) or, it can also be a pre-trained set of values inherited from another task. **W** gets updated with every new data instance with the objective of maximizing the likelihood of the instance belonging to one of the two classes. A function, like *cross-entropy*, can be used for this parameter estimation process. The cross-entropy function compares each of the predicted probabilities to actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value to get an estimate on how close (or far) are is the classifier from the actual value:

$$\frac{1}{N} \sum_{i=1}^{N} -(y_i * log(p(x_i)) + (1 - y_i)\ log(1 - p(x_i))) \tag{2.5}$$

where, y_i are the actual outputs and $p(x_i)$ are the predicted outputs. Note that the function, by its design, pushes the prediction estimation towards one of the classes. This resultant value can also be called as *loss* or *error* and consequently, such functions are typically referred as *loss functions*. This computed loss is used to update the weight parameter matrix $\mathbf{W}$ iteratively, till the classifier reaches convergence i.e., the loss becomes 0 or falls within a predefined threshold close to 0.

Once the classifier is successfully trained, it can be used for making categorical predictions. New data instances, within the scope of the classification problem being addressed, can now be provided as input to the classifier and the classifier in turn predicts a class as an output. The success of a supervised classifier largely depends on the data instances on which it has been trained upon. If the number of data instances (representing classes) are imbalanced, then the learning can be biased in the sense that the majority class can influence generalized learning – which is desirable in such modelling setups. The learning can also be hindered if the classifiers are over-trained or under-trained (Chakraborty et al., 1995).

Note that there are many other classifiers which can be used to address classification – each representing a different algorithm to maximize the probability of assigning a data instance to a class. For instance, 1) In the example above, the Logistic Regression classifier can be converted to a neural network by the introduction of hidden layers which also help in addressing non-linearity more effectively (Glonek and McCullagh, 1995); 2) *k-Nearest Neighbours* assigns objects to the class that most of its nearest neighbours in the multidimensional feature space belong to. The number k is the number of neighbouring objects in the feature space that are compared with the classified object (Kramer, 2013); 3) *Support Vector Machines* (SVMs) use a hyperplane in an N- dimensional space to classify the data points, where N (as above) represents the number of variable parameters. The hyperplane in an SVM is selected (from potentially many others) based on the the fact that it represents the largest margin (separation) between the two classes. Essentially, the hyperplane is chosen so that the distance from

it to the nearest data point on each side is maximized (Noble, 2006).

2.3 Clustering

Clustering (also known as *cluster analysis*) is the task of categorizing objects such that they get organized into different groupings, called *clusters or clustering solutions*, where the objects in one cluster should be as similar as possible to each other and dissimilar from the rest of the objects in other clusters (Nagy, 1968). If every object is exclusive to one cluster, then its called *hard* clustering. And, if an object has a variable degree of membership in each of the output clusters, then its called *soft or fuzzy* clustering (Ruspini, 1969). The objects are data points represented as (feature) vectors in a multidimensional space and, the similarity or dissimilarity (distance) is an outcome of the differences between the latent features of the vectors (Jain et al., 1999).

Clustering is an *unsupervised classification* technique because its objective is the classification of unlabelled data into meaningful groups *without* the use of any pre-classified (training) data. Subsequently, it is also a popular tool for exploratory data analysis when preliminary assessments have to be made about the relationships between data with minimal assumptions and little prior information (about the data) (Jain and Dubes, 1988). On this point, it is worth noting that this lack of prior information is sometimes problematic because, within the clustering solutions generated by a clustering algorithm, it is difficult to ascertain which clusters are a product of the structure of the data and which (if at all) are an artefact of the method used. If the latter is *true*, it leads to a *misinterpretation* of the structure of the data, which is the exact opposite to the purpose of clustering. Therefore, the validation of an overall optimal solution *often* requires human supervision by the way of visual inspection(s) (Everitt et al., 2011). *Principal Component Analysis* (PCA) [8] is used to project the higher dimensional data

[8] PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data

into a lower dimensional space, generally for better visual interpretation of higher dimensional interactions.

Starting from the observation that clustering is a type of grouping, the notion of *what constitutes a cluster* is not easy to define. This is largely due to the fact that: 1) there are many types of clustering algorithms (discussed next), each addressing a different problem and consequently presenting a unique structural framework with respect to what clusters are; 2) depending on the applicability, the choice of algorithmic parameters vary. For instance, in distributional semantics Cosine or Euclidean *distance* [9] is an effective clustering distance metric (Lin et al., 2013). On the other hand, graph based applications might find *graph edit distance* [10] (Gao et al., 2010) or *geodesic distance* [11] (Tekir et al., 2011) to be more useful; and, 3) clustering as been applied to divergent disciplines, such as linguistics (Täckström et al., 2012), biomedicine (Wiwie et al., 2015), image processing (Dhanachandra and Chanu, 2017), geography and culture (Breschi and Malerba, 2001; Taras et al., 2016), weather forecasting (Chakraborty et al., 2014), archaeology (Baxter, 2015) and marketing (Seret et al., 2014) (amongst many others), thus, implying the use of datasets with very different properties.

There are approaches that attempt to present a unified theory of characterizing clusters and cluster quality, by defining *axiomatic frameworks* which do not depend on clustering algorithms, parameters and data specificity, thus, circumnavigating the issues mentioned above (Ben-David and Ackerman, 2009; Zadeh and Ben-David, 2012). Nonetheless, in general contexts, clustering mechanisms based on homogeneity (internal cohesion) or separation (external isolation) of objects are still considered to be sufficient and perhaps the most straightforward approach (Kleinberg, 2003).

comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on (Jolliffe, 2011).

[9] A function over a set of elements that defines their distance from each other in an n-dimensional hyperplane.

[10] A measure of similarity (or dissimilarity) between two graphs which computes the minimum number of graph edit operations required to transform one graph to another.

[11] The total number of edges in the shortest path between two vertices.

2.3.1 Types of Clustering

Clustering algorithms can be divided into two categories at the top level: 1) Partitional; and, 2) Hierarchical.

Partitional algorithms group objects into sets of disjoint clusters where the number of output clusters is typically predefined by the user as a design decision (for details see Dubes, 1987). The algorithms falling under this category produce output clusters by optimizing over a similarity or a distance based objective function. The optimization can be done over all the objects, globally or on a subset of the objects, locally. However, its difficult for the function to optimize over large datasets at one go. Therefore, such algorithms are typically iterative in nature with different starting states for each iteration. And, the iteration with the best configuration is used for output clustering.

Hierarchical algorithms create a series of partitions which typically run from a single cluster containing all objects to n clusters, each containing a single object (Johnson, 1967). The end result is visually akin to a *dendogram* that represents a series of nested clusters. This dendogram can be (conceptually) broken at different levels to reveal different types of similarity (or dissimilarity) based, inter-cluster and intra-cluster associations. These algorithms can be either *agglomerative* or *divisive*.

The agglomerative approach implies a 'bottom-up' clustering. At initialization, a proximity matrix is computed which contains the distance between each pair of patterns and each data object represents a cluster i.e. $\mathtt{M}$ objects $= \mathtt{M}$ clusters. Then, a merge operation is performed which fuses the most similar pair of clusters into a larger cluster and the proximity matrix gets updated to reflect this merge operation. The algorithm iterates till a single cluster containing all the objects is obtained. The divisive approach is the exact opposite of the former: a large group of objects is partitioned till singleton clusters emerge.

Hierarchical approaches suffer from the disadvantage of not being able to revert back on the decisions made in the previous steps. The decision to

fuse two clusters into one or divide a cluster into two is irrevocable and if, in theory, a solution is found to be suboptimal at later stages then nothing can be done about it (Kaufmann and Rousseeuw, 1990).

Both partitional and hierarchical approaches have multiple offshoots, where the algorithmic innovation varies according to the problem but similarity and dissimilarity based clustering criteria remain a central theme. A detailed description of these approaches can be found in surveys done by Jain et al. (1999), Xu and Wunsch (2005) and Aggarwal and Zhai (2012).

2.3.2 K-means Clustering

K-means clustering is a popular partitioning algorithm which has been in use for over four decades[12] (Everitt et al., 2011). It aims to partition `M` objects, in an `N` dimensional space, into `K` clusters. Each of the `M` objects belongs in the cluster it is nearest to, based on its distance from the `N` dimensional prototypical vector of each of the `K` clusters. This prototypical vector is computed by calculating the mean of all the objects within that cluster (MacQueen et al., 1967).

One disadvantage is that the algorithm is limited in the way of specifying a static number of clusters (`K`) at initialization. Another issue is the potential non-convergence of the objective function over large datasets. Lloyd (1982) proposes an algorithm that computes the local optimum (but not necessarily global) for this problem but without a guarantee that the local optimum is in fact the optimal solution. Plausibly because as shown by Dasgupta (2008), the K-means algorithm is computationally difficult (NP-hard[13]) but applying specific heuristics can make it converge quickly to a local optimum (Arthur and Vassilvitskii, 2006; Ostrovsky et al., 2013).

The algorithm is iterative and initializes with a random set of `K` observations from the dataset, initially used as means for the `K` clusters. There are

[12] And, found to be the most relevant approach for exploratory data analysis in this thesis

[13] NP (or Non-deterministic Polynomial) hard problems are those problems which can be *verified* in polynomial time i.e. given a solution to a problem, it can be validated for correctness in polynomial time. However, the algorithm in place does not guarantee to arrive at a *solution* in polynomial time. For more information, refer to Woeginger (2003)

many other techniques to initialize K-means clusters (Celebi et al., 2013), but it has been observed that random seeding along with fine tuning of model parameters works well for most cases. Post initialization, there are two iterative steps: 1) Each object is assigned to one of the K clusters whose mean has the least squared euclidean distance from that object; and, 2) Next, for each of the K clusters, new means (centroids) are computed from the revised clusters.

$$\sum_{i=1}^{k} \sum_{j=1}^{n} (\| x_i - v_j \|)^2 \; = \; 1 \tag{2.6}$$

The objective function that the algorithm aims to minimize is the squared error function, as defined in equation 2.6; where, $(\| x_i - v_j \|)$ is the Euclidean distance between a point, x_i and centroid v_j, iterated over all k points in the i^{th} cluster, for all n clusters. The algorithm is said to reach a convergence (local optima) when the assignments of objects to clusters no longer change.

2.4 Knowledge bases

A *Knowledge Base* (*KB*) is an information resource which stores facts in a structured manner (Feigenbaum, 1977) related to specific domains like, linguistics (Miller, 1995, WordNet), bio-medicine (Bodenreider, 2004, UMLS), natural sciences (Rich and Venkatasubramanian, 1987, MODEX) and organizations (Bhatt, 2001; Aitken et al., 2010, PCF-APQC) or more generally, the real world (Bollacker et al., 2008, Freebase).

The *facts* are a description of interactions between entities (concepts and instances) or their attributes through relations between them (MacGregor and Burstein, 1991). That is to say, pairs of (*entity, entity*) or (*entity, attribute*) or (*attribute, attribute*) might be connected through relations which explain the associations between the paired elements; each association construing a fact. These facts are also termed as *acquired* knowledge,

in the sense, that they are typically extracted and organized with the help of domain experts (Davis, 1979) or automatically from already available domain resources (Kononenko, 1990) or social interactions (Carley, 1986).

The term *knowledge base* was coined to distinguish this storage framework from other popular data storage frameworks, such as *hierarchical or relational* data management frameworks. The key difference is that KBs store information in a structured form similar to the way humans perceive information, therefore, giving the ability to design applications which process information to *knowledge* on the lines of human reasoning. Whereas the other frameworks tend to store data in flat structures (called *tables*) which are meant for efficient information storage and retrieval (Brodie and Mylopoulos, 1986).

The first ever usage of KBs is seen within closed-domain applications, i.e. systems which required only domain knowledge within restricted context(s). These applications are referred to as *expert systems*[14] which are used solve complex problems by reasoning through bodies of knowledge that can be created and maintained by domain experts. For example, GPS – a program that generates transitional solution states starting from a problem and leading to a solution (Newell and Shaw, 1959); DENDRAL – a program to determine the molecular structure of an unknown chemical compound (Buchanan et al., 1969); and, MYCIN – a program to advise physicians regarding antimicrobial therapy selection (Shortliffe, 1974).

However, KBs have evolved from storing closed domain knowledge towards storing generic world-knowledge (Bollacker et al., 2008) due to factors like increasingly sophisticated algorithms, software and hardware. Additionally, the internet has made large amounts of globally distributed information available at fingertips. Therefore, in the recent past, manual creation and curation of general purpose KBs has become a seemingly quixotic undertaking.

[14] Expert Systems are considered to be amongst the first truly successful artificially intelligent software. These are divided into two subsystems: an inference engine and a knowledge base. The inference engine performs the deductions on the knowledge (and rules) contained within the knowledge base. (Lucas and Van Der Gaag, 1991)

A popular alternate to these human-intensive endeavours is a combination of collaborative efforts between various communities (industry experts, academicians and general public) and (semi) automatic structured information extraction from unstructured resources. The resultant *large-scale* KBs are also nowadays called *Collaboratively Constructed Knowledge Bases* (*CCKBs*), like, Freebase (Bollacker et al., 2008), DBPedia (Bizer et al., 2009), WikiData (Vrandečić and Krötzsch, 2014), and Yago (Rebele et al., 2016).

Such CCKBs have come to play a prominent role in building semantic information systems, specially in the CL and NLP communities, forming the basis for a wide range of applications (Hovy et al., 2013) such as semantic inferencing (Haarslev and Möller, 2003), question-answering (Berant et al., 2013; Krishnamurthy and Mitchell, 2015), semantic parsing (Yih et al., 2015), named-entity representation (Bordes et al., 2011; Toutanova et al., 2015), entity linking (Shen et al., 2012) and entity typing (Yaghoobzadeh et al., 2018).

A consequence of creating large-scale CCKBs is a constant need for their curation with attention to scalability and robustness in terms of information extension, performance and memory usage[15]; which has been a widely researched problem (Speel et al., 1995; Jarrar and Meersman, 2002; Nakashole et al., 2011).

There has been considerable effort put into developing effective structural and semantic frameworks, such as Resource Description Framework (RDF) (Brickley et al., 1999) and Web Ontology Language (OWL) (Bechhofer et al., 2004), respectively. RDF defines the structure of the data. In RDF, a *fact* can be represented as three-element-tuple (triple) of *subject : predicate : object* elements where the *predicate* is the relation and *subject : object* are entities or their attributes or attribute-value literals (a number or a string). OWL, while also expressed in triples, adds semantics to the

[15]To address performance and memory issues, the KB community is also invested in developing state-of-the-art benchmarks to optimize the technologies built on the above mentioned frameworks (Guo et al., 2005; Lopez et al., 2013; Aluç et al., 2014).

underlying schema developed through RDF. Simply put, RDF defines *how* to write and OWL defines the semantic validity of *what* can be written.

2.4.1 Incompleteness and Knowledge Base Completion

Issues of knowledge extension also bring forward a crucial shortcoming of CCKBs, namely their *incompleteness* (Min et al., 2013; West et al., 2014) – not just with respect to the entities that they cover, but also with respect to their attributes that are either nominally covered or seem to get added over time. This is not surprising: when a contributor to a knowledge base adds an entity, they will probably concentrate on the most salient attributes (e.g., for a scientist, *field* or *affiliation*), while other attributes (such as *parents* or *place of birth*) may be added later or never. This realization has led to a large boost to work in the area of *Knowledge Base Completion* (*KBC*), that is, the prediction of attributes of entities that are currently missing from the CCKBs (Bordes et al., 2013; Socher et al., 2013a; Min et al., 2013; Guu et al., 2015).

Knowledge Base Completion (KBC) as a task, has its origins in the Information Extraction (IE) domain as a Relation Extraction (RE) task used to populate KBs. Traditional IE systems perform RE by searching for *relation-specific* patterns based on hand- labelled data and relation templates or by (semi) supervised machine learning in which training samples are extracted through templates or sequence labelling (Brin, 1998; Agichtein and Gravano, 2000; Nahm and Mooney, 2000; Culotta et al., 2006). While the hand-labelled methods are labour intensive and scale linearly with the number of relations to extract, the statistical methods produce moderate precision and low recall while being memory intensive. With the increased use of unlexicalized parsers (Klein and Manning, 2003), the extraction paradigm moved towards *relation-independent* Open Information Extraction (OIE) where relations are extracted automatically over large corpora in the form of *(entity1, relation, entity2)* tuples with higher accuracies (Yates et al., 2007). However, these methods falls short

on capturing complex linguistic patterns and domain specific ontological structures. One effective approach, amongst others, has been to extract semantic knowledge from general corpora with the aid of RDF, in conjuction with OWL, to precisely associate extracted patterns to the KB framework (Buitelaar and Cimiano, 2008). Thus, bridging together the elements of KBC modelling as we know it today.

The initial use of distributional semantics for Knowledge Base Completion can be seen in the work by Bordes et al. (2011), modelled as a task of supervised Freebase and Wordnet relation prediction from neural embeddings constructed using CCKBs as a source. Freitas et al. (2012) discuss the applicability of using distributional representations for question-answering systems on large-scale common-sense knowledge bases. Socher et al. (2013a) subsequently show that performance of relational querying models improves when entities are represented as an average of their constituting word representations built from unsupervised large-scale corpora. Bordes et al. (2013) use such distributional representations to identify the connectivity patterns between entities in a multi-relational learning setup on knowledge bases to link entities and predict new relations, while Freitas et al. (2014) introduce a complementary distributional semantic layer to cope with semantic approximation and incompleteness of common-sense information. Toutanova et al. (2015) jointly learn continuous representations for entities and textual relations, Basile et al. (2016) populate object-location relations in a knowledge-base by computing the prototypicality between objects and locations extracted from text, Nickel et al. (2016) use distributional representations to generate optimized Holographic embeddings to generate compositional embeddings of binary relational data and, Trouillon et al. (2017) demonstrate that complex embeddings are more optimized as compared to real valued embeddings for link prediction. Overall, the focus has been on constructing optimized embeddings for task specific application where *Link prediction* – connecting two entities through a binary relation, is a popular evaluation setup. The underlying objective for all of the above is to find expressive yet generalized representations which can

cope with the ever increasing scalability challenges within the scope of this research area.

The application of distributional methods in KBC has been multi-faceted and well documented. However, there are certain under-investigated aspects in these studies that remain to be answered: there is little information on the kinds of relations, entities or attributes that are easy or difficult to predict. And, what are the reasons arising through the distributional frameworks (if at all), that make such predictions easy or difficult.

2.5 Named Entity Recognition and Classification

Named Entity Recognition and Classification (NERC) deals with the objective of recognizing the occurrence of Named Entities in unstructured text followed by their classification into various class hierarchies (Nadeau and Sekine, 2007).

Named Entities (NEs), a term coined by Grishman and Sundheim (1996), are entities which in language are referred to through the use of one or more rigid designators (Kripke, 1982). The designators are usually proper nouns that help linking a real-world referent as an instance of a class (i.e., a concept) (Quirk, 2010). For example, *Abraham Lincoln* can be considered as an NE which belongs to the class *person*; in other words, the two are in a semantic relation: $NE\ (is\_a/type\_of)\ class$.

The beginnings of NERC as a task can be traced to the domain of Information Extraction where there was a need of recognition, extraction and use of structured knowledge (from linguistic expressions) within the business and defence domains for reasoning oriented NLP applications. This structured knowledge is heavily loaded with NEs and recognizing them in unstructured text is a challenge because they occur in varied contexts, for example, *Lincoln was the US president* and *For 1923, Lincoln Motors produced over 7000 cars*. After the NEs have been recognized in text,

to use them as semantically relevant informational units, it is essential to identify the classes they belong to: *Lincoln – person* and *Lincoln Motors – organization* (MacGregor and Brill, 1992). Thus, NERC conceptualized but, originally as two independent sub-tasks: Named Entity Recognition (NER) and Named Entity Classification (NEC). This is because NER was considered a sequence labelling task through pattern-matching with syntactic (segmentation, chunking and parsing) undertones (Palmer and Day, 1997). And, given that the NEs are already identified, NEC was then treated mainly as a classification problem; solved via heuristics or statistical machine-learning (Collins and Singer, 1999). At this point, note that we consider classification after recognition to be implicit and therefore, we do not bifurcate into NER and NEC in what follows.

2.5.1 NERC Class Hierarchies

One of the earlier works in NERC involved extracting company names from financial news (Rau, 1991) and, in its nascent stages, NERC was limited to recognizing proper nouns in text for a single domain. However, with the growing popularity of NERC, the Message Understanding Conference (MUC) (Grishman and Sundheim, 1996) laid the cornerstone in the development of class hierarchies by being the first to categorize NEs to seven classes: *organization, location, person, date, time, money and percent expressions*. However, this flat structure soon began transforming into a hierarchical structure to accommodate the requirements brought on by the inclusion of different domains, languages and extraction methodologies. For instance: 1) an entirely new class *artifact* got added to process Japanese newswire text (Sekine and Isahara, 2000); 2) *location* got divided into eight finer subsets *country, state, city, etc.* to explore the effectiveness of automated systems extracting fine-grained information (Fleischman, 2001); and, 3) contrastively, *location and organization* were subsumed into a new coarser class *facility* for a multi-lingual multi-domain corpus (Doddington et al., 2004).

The increasing prominence of NERC in Natural Language Processing[16] brought about a systematic extension of class hierarchies to address the needs of open-domain frameworks. Sekine et al. (2002) created a hierarchical resource containing 150 generic classes which was subsequently refined to 200 (Sekine and Nobata, 2004). This also brought about standardization of datasets through shared tasks (Tjong Kim Sang and De Meulder, 2003; Kartsaklis et al., 2013; Levy and Goldberg, 2014b). And these resources (being hand-crafted, thus, very accurate) are still used to develop state-of-the-art NERC approaches (Lample et al., 2016; Ma and Hovy, 2016).

A parallel trend in the recent past, as compared to the hand-crafted resources above, has been to extract knowledge-base level *fine-grained* class hierarchies automatically, in both supervised and unsupervised setups, through open-domain knowledge sources like Freebase (Ling and Weld, 2012a, extract 112 fine-grained classes), OntoNotes (Gillick et al., 2014, introduce context-dependent 12017 NE typed mentions), Yago (Yosef et al., 2012, extract 505 classes) and *Balanced Corpus of Contemporary Written Japanese (BCCWJ)* (Iwakura et al., 2016, extract 2, 464 NE types). Since modern day knowledge-bases incorporate ontological information about NEs, these are also directly used to test newer approaches of fine-grained NERC (Mai et al., 2018).

2.5.2 Traditional NERC Frameworks

Traditional NERC systems are based on hand-crafted rules for both entity recognition and classification. These systems require machine readable textual annotations. The Standard Generalized Markup Language (SGML) (Goldfarb, 1990) is used by Grishman and Sundheim (1996) to create annotation tags (ENAMEX, TIMEX and NUMEX) which are used to mark

[16]Conferences like European Chapter of the Association of Computational Linguistics (EACL) 1999, Empirical Methods in Natural Language Processing (EMNLP) 2000 and Conference on Computational Natural Language Learning (CoNLL) 2002-2003, began staging NERC main track sessions and workshops.

NE occurrences. ENAMEX indicates information about *persons, locations and organizations*, TIMEX about *date and time* and, NUMEX about *money and percentages*. Additionally, NE lists, called Gazetteers, are also heavily used for pattern matching (Krupka, 1995). The effectiveness of gazetteers, however, is questionable because although their size can grow in proportion to the domain scope (a system for a company vs. open domain), language (resource rich vs. resource poor) and the variety of linguistic expressions with which NEs are referred to in text (Cucchiarelli et al., 1998) but, it has been observed that large gazetteers do not necessarily contribute to higher accuracies (Krupka and Hausman, 1998). In fact, Mikheev et al. (1999) show that rule-based systems which use partial probabilistic mapping with minimalistic gazetteers (only common and well known NEs, like location names) perform at par with systems using large gazetteers.

Similar efforts (Borthwick et al., 1998; Borthwick and Grishman, 1999), that reduce the use of additional lexical resources, popularized *supervised learning* as the preferred technique for NERC (Tjong Kim Sang and De Meulder, 2003, and references therein). Supervised learning requires the creation of training data which is made from both positive and negative examples. Ideally, the examples should be diverse enough, in terms of word-level features and rules, to sufficiently capture all the NE instances of a given type. Statistical sequence labelling and classification algorithms can then use this training data for NERC over seen or unseen text (McCallum and Li, 2003; Culotta et al., 2005). However, supervised techniques require the creation of large amounts of labelled training examples for accurate results, which may not always be possible (Collins and Singer, 1999).

One suitable alternate in such eventualities is *semi-supervised learning*; also called *weak supervision*. In semi-supervised setups, a much smaller amount of labelled data (as compared to supervised setups) is used as starting *seeds* from which systems extract *contexts* which in turn are used for recognition and classification of a much larger set of NE instances (Liang, 2005). While the seeds are typically NE expressions, the contexts can

be collected in different ways from text, for example, through regular expressions (Brin, 1998), spelling rules (Collins and Singer, 1999), syntactic relations (subject-object) (Cucchiarelli and Velardi, 2001) or even distributional similarities (Pasca et al., 2006). Semi-supervised methods greatly reduce the burden of creating and curating copious amounts of labelled data. But, the technique works best when the unlabelled documents used for seeding contexts are rich in NEs and coreferences (Ji and Grishman, 2006). Another alternate to supervision is to use *unsupervised learning*. Techniques like context based *clustering* can be used for NE recognition (Lin and Wu, 2009). For NE classification, first the NE classes can be identified from: 1) existing lexical resources (WordNet) (Alfonseca and Manandhar, 2002a); 2) contexts or patterns extracted from documents (Evans and Street, 2003); or, 3) lexical co-occurrence statistics (Etzioni et al., 2005). The subsequent classification can also be unsupervised, but due to the general non-guarantee at reaching an optimal solution (see, Section 2.3.1), one can always use (semi) supervised or rule based approaches as well.

2.5.3 Distributional NERC Frameworks

The manual or semi-automatic feature extraction processes (at word, document and corpus level) for NE expressions, for both supervised or unsupervised setups, is a tedious process (Leser and Hakenberg, 2005; Nadeau and Sekine, 2007). This problem is alleviated with the use of distributional representations created from co-occurrence information that is extracted from corpora in an unsupervised manner, see Section 2.1.2. These representations require minimal feature engineering but excel at capturing syntactic and semantic features of linguistic units (Bengio et al., 2013). In the case of NERC, as it with and many other tasks, the combination of non-linear models using distributional representations as input are known to perform at par with the other state-of-the-art non-distributional benchmarks (Collobert et al., 2011). The application of distributional representations

makes NERC a three-staged task with minimal human intervention due to their *input–encoder–decoder* frameworks.

In the first stage, distributional representations are created which are supplied as input to the second stage. The representations can be created at word-level or at character-level. The former have been discussed in detail in Section 2.1.1. The latter representations (Zhang et al., 2015; Ma and Hovy, 2016), have been found useful for exploiting explicit sub-word-level (prefix and suffix) information and for sharing information of morpheme-level regularities. Another big advantage is that character-level embeddings allow for the construction of unseen words. Lately, in addition to the generic word and character level representations, there are context and task specific hybrid embeddings created by adding information from: lexical features (Ghaddar and Langlais, 2018), syntax (Jie and Lu, 2019), auxiliary structure-based representations (Devlin et al., 2019) or multi-modal data (Lu et al., 2018) to the corpus evidence.

In the second stage, the distributional representations of the NE expression are given as input to an *encoder* like a Convolutional Neural Network (CNN) (Collobert et al., 2011; Strubell et al., 2017), Recursive Neural Network (RNN) (Li et al., 2017) or variants of Recurrent Neural Network (RNN), for example: Long Short Term Memory (LSTM) and stacked LSTMs (Lample et al., 2016) or Bi-directional LSTMs (Ma and Hovy, 2016), so that semantically expressive context infused encodings can be generated. A relatively new approach are the Transformer based models (Vaswani et al., 2017) that represent a new paradigm in NERC due to the fact that the representations are contextualized and can be further fine tuned for specific tasks. Significant increase in performance have been observed with the use of pretrained Bi-directional transformer models (Baevski et al., 2019) which are almost equivalent to the current state-of-the-art by Devlin et al. (2019).

The third, and the final, stage is where the context based representations are given as input to a *decoder* to produce a sequence of *tags* (classes)

corresponding to the input sequence. Cui and Zhang (2019) use a multi-layer perceptron (MLP) with Softmax as a decoder, converting NERC as a multi-class classification problem where the tag of each word is predicted without taking the neighbours into account. Conditional random field (CRF) based decoders (Peters et al., 2018) are also a popular choice when non-contextualized representations, such as Word2vec (Mikolov et al., 2013a) or GloVe (Pennington et al., 2014), are used. Another option are the Recurrent Neural Network (RNN) based decoders which are generally considered faster than CRFs but due to their dependency on the previous output step, these might lead to inefficiency and problems of parallelization (Shen et al., 2017).

In addition to the work above, most of the other recent NERC indicate that the application of distributional approaches in NERC goes hand-in-hand with the increasing need (and experimentation) of fine-grained classification of NEs (Shimaoka et al., 2017; Abhishek and Awekar, 2017; Choi et al., 2018; Obeidat et al., 2019). And, *how* fine-grained can the classification get is ultimately dependent on the task at hand (Fisher and Vlachos, 2019). But, little has been done in terms of empirically understanding the potential and limitations of using distributional representations for NE classifications on the linguistic side.

Overall, with the advancements in data representation frameworks and machine-learning technologies, NERC continues to remain an active and a steadily growing research area over the last two decades. The attention is due to the fact that it is considered as a pre-processing task to other downstream applications (such as relation prediction, question answering, machine translation, etc.). Thus, as the applications evolve, so does NERC.

Chapter 3

Instantiation - Part I

In Chapter 1, we touched upon the subject of lexical relation of entities remaining distributionally under-addressed. We postulate that this is because there is an assumption that conclusions over phenomena related to lexical generalizations will also hold for entity related phenomena (Section 1.1.2). The implication being that distributional representations of entities and categories are considered *at par* or that at-the-least, their behavioural differences, if any, remain unrealized.

In this chapter, we address our first research question (as proposed in Section 1.3): *Can we model the lexical relation – Instantiation (as defined below) – distributionally?* Our experiments and analyses test the hypotheses that Instantiation can be modelled distributionally and that entities and category representations are fundamentally distinct from one another. The ultimate research goal being the advancement of understanding of how entities and categories are mirrored in distributional patterns.

3.1 Background

We define **Instantiation** as a lexical relation between two words (specifically, *lemmas* that are not in context) where the first word denotes an entity (a proper noun) and the second word denotes its instantiating category (a common noun). The instantiation relationship between the entity and the category holds if the entity belongs to that category; in other words, the entity instance satisfies the concept represented by the category (MacGregor and Brill, 1992). It is the most prominent relation between an entity and a

category whereby a referent gets connected to a concept. Linguistically, in its simplest form, instantiation occurs as a copular relation: in English, for example, instantiation can be expressed by *Germany is a country*; where *Germany* is the entity and *country* is the category.

Instantiation remains under-addressed in formal as well as in computational semantics. In formal semantics, instantiation is assumed rather than learned. For example, Heim and Kratzer (1998), present a systematic development of a semantic theory for natural language based on the principles of Frege's treatment of semantic composition as a functional application[1]. To this purpose, they provide a lexicon that specifies the denotation of non-functional elements of a language including: 1) entities (specifically, individuals) as – ***[Abraham Lincoln]*** = *Abraham Lincoln*, where the italicised term denotes the referent; and, 2) categories as a set of individuals with pre-computed truth-conditions in relation to that category – **[president]** $=$ $[Abraham\ Lincoln \rightarrow 1, Harry\ Potter \rightarrow 0, ...]$. Thus, instantiation is not learned but provided by the way of definition.

In computational semantics, considerable focus has been put towards understanding the semantics of categories (represented by common nouns, adjectives or other word classes) with respect to words and their lexical relations (Cruse et al., 1986; Geeraerts, 2010, and Section 1.1.2). A plausible motivation comes from formal semantics, which has dedicatedly investigated categories with respect to their vagueness and gradability (Klein, 1980; Lakoff, 1973; Tversky and Gati, 1978). A category can be vague if the membership of an entity in that category cannot be determined (due to incomplete or partial contextual information). For example: 1) a person can be *tall*, *not tall* or somewhere in-between; and, 2) *that robin is more of a bird than an ostrich*. Gradability implies that a category is compatible with comparatives, equatives, superlatives and other degree modifiers. For example: 1) *taller, tallest or more | less | equally tall*; and, 2) *X [is similar*

[1] Frege's principle of compositionality states that the meaning of a complete sentence must be explained in terms of the meanings of its subsentential parts, including those of its singular terms (Mitchell and Lapata, 2010), i.e. the meaning of a sentence is a function of the meaning of its smaller parts.

to | virtually] Y. Note that where common noun categories are concerned, there are two schools of thought. The classical view (Katz and Fodor, 1963), promotes that these are completely semantically interpretable. That is to say that common noun categories are neither vague nor gradable and, there exists a set of necessary and sufficient conditions for categorization (membership in their denotation). This view is rejected on philosophical and empirical evidence which shows that common noun categories are inherently vague and gradable (Rosch, 1975; Armstrong et al., 1983); in other words, a member of a category may share a slightly different set of properties with every other member of that category (Sassoon, 2013). In this study, we do not follow up on this motivation.

Distributional semantics, one of the prominent frameworks in computational semantics, too has focussed on category based lexical relations (like *synonymy, antonymy, hypernymy, hyponymy*; see Section 1.1.1) These relations imply varying degrees of gradability between their words. Given the methodology of meaning construction by distributional models (in Section 2.1.1), distributional semantics has proven to be an effective framework for lexical relational modelling. The focus, however, has been on examining gradability via semantic similarity and relatedness; such as the fact that cats are animals (Baroni et al., 2012), similar to dogs (Landauer and Dumais, 1997), and shed fur (Erk et al., 2010). On the other hand, the distributional investigation of entities and their lexical phenomena (like instantiation) remains largely un-investigated.

In contrast, instantiation has been addressed by the Information Extraction (IE) and Knowledge Representation (KR) communities, although from an applied perspective. These communities are interested in extracting and modelling all types of real-world knowledge and consequently, instantiation (or more generally, categorization of entities) is a prominent sub-task. In IE, categorization of entities falls under the purview of Named Entity Recognition and Classification (NERC) and in KR it is termed as Named Entity Typing (NET). Despite similar goals i.e., categorization with high accuracies, these tasks differ from each other in their formulation and

modelling approaches. We discuss these and compare them with our work in greater detail in the next sections. At this point, its suffice to say that at a broad level our work differs from the above because our focus goes beyond categorization. We are more focussed on examining entities, categories and instantiation with respect to their distributional behaviour and geometry in a semantic space.

Plan of the Chapter: In Section 3.2, we discuss previous works which investigate the distinction between entities and categories and, have commonalities with our research aims of modelling instantiation. Section 3.3 is a detailed description of the data that we use for our subsequent experiments and analyses. In Section 3.4 we perform exploratory data analysis on the distributional geometry of entities and categories within a distributional space. And finally in Section 3.5, on the basis of our main hypothesis of this chapter, we model instantiation distributionally as a classification task and discuss the outcome and analyses with respect to the hypothesis.

3.2 Related Work

In the next sub-section, we discuss the previous work in computational semantics related to entities and categories. Through this discussion we aim to highlight how our focus is on those aspects which have either previously not been dealt with or are complimentary to our objectives.

3.2.1 Entities and Categories

When it comes to examining entities (that are instances) and categories (that represent concepts), the studies are relatively few and sparse.

Alfonseca and Manandhar (2002b) distinguish entities and categories in WordNet (Fellbaum, 1998) annotating synsets manually as well as automatically. They manually annotate 51,553 categories and 7033 entities where a synset and a leaf-synset is tagged as a category if it has immediate

hyponymys and hypernyms, respectively. The remaining leaf-synsets are tagged as entities. To automatically model this distinction, they train a Maximum Entropy classifier to identify instances based on manually selected features like, precedence of determiners to words in the synsets, capitalization of words, plurality, etc. On a randomly selected subset of 300 categories and 150 entities, they report an accuracy of 96.62% which suggests that it is relatively straightforward to distinguish words as entities and categories based on orthographic and syntactic information. However, they do not report results on test data that is not observed in training data i.e., there is no analysis on the generalization capabilities of their classifier on unseen (or new) data.

Miller and Hristea (2006) *formally* incorporate this distinction between entities and categories in WordNet. They choose a manual approach in order to reach (near) perfect accuracy. Synsets are tagged as entities based on three characteristics: 1) the candidates are nouns; 2) the candidates are proper nouns (hence, they are capitalized); and, 3) the synsets should not have hyponyms i.e., the words in a synset have a unique referent. The last one being the most important. From a set of 24,073 potential candidates, 7671 synsets are tagged as entities where the agreement coefficient kappa is 0.75; the high agreement indicating substantial correspondence between the authors, if not perfect[2]. Thereon, WordNet has become a standard go-to lexical resource for research communities and NLP applications that require (or make use of) this distinction.

Building on the work of Miller and Hristea (2006), one can see more efforts towards automatic differentiation between entities and categories in large scale linguistic resources and corpora. For example, Zirn et al. (2008) automatically differentiate them in a large scale taxonomy built from Wikipedia. They tag words as entities based on a combination of outputs from a Named Entity Recognizer as well as the orthographic and surface-level syntactic features mentioned above. On the other hand, categories are tagged based on position in the taxonomy i.e., a candidate is a category if it

[2]For details on computing agreement statistics see Siegel (1956).

has two direct hyponymys or one direct hyponym that is also a hypernym. They identify 15,472 entities and 111,652 categories with an accuracy of 84.52%. Reiter and Frank (2010) identify generic noun phrases, involving common noun categories, from a corpus (ACE-2 by Mitchell et al. (2003)) using a supervised Bayesian classifier. They use a combination of syntactic (number, person, word-type, part-of-speech, dependency relation, etc.) and semantic (word sense classes, sense granularity, etc.) features to learn the classification. On their balanced dataset of 10,000 generic phrases (containing categories) and an approximately equal number of non-generic phrases (containing entities), they report an accuracy of 83.7%. The classifier can be subsequently extended to distinguishing categories from entities.

The underlying theme of these works is to provide lexical resources which carry the demarcation between entities and categories which in turn can be used by data driven frameworks that make use of this distinction. These efforts are complimentary to our work since we work with entities and categories that have already been identified. Moreover, the main focus of these works is neither to investigate the criteria under which entities are instantiated by their categories nor to analyze the distinctions between them. We, on the other hand, address the two in a distributional setup.

Additionally, there is also considerable work in identifying predefined semantic relations (that typically occur as predicate-argument structures) which include both entities and categories (Hendrickx et al., 2010, and subsequent works). However, in contrast to our work which is primarily based on entities denoted by proper nouns, these works typically use private or unnamed entities. Examples of such relations and their lexical patterns are: *entity – origin* (the letters from foreign countries), *entity – destination* (the boy went to bed) and *member – collection* (my apartment has a large kitchen). Such relations, in principle, are not comparable to instantiation (a lexical relation in the classical sense) and are more beneficial for downstream tasks of knowledge prediction and representation like common-sense reasoning (Socher et al., 2013a; Freitas et al., 2014).

3.2.1.1 In Distributional Semantics

In distributional semantics as well, we see less work specifically on the distributional representation of entities. We list out some previous work which highlights the ways in which entities and categories have been employed in distributional semantics.

Lewis and Steedman (2013) build a semantic parser by mapping language to first-order logic representations capable of capturing the meaning of function words. To this purpose, they use distributional representations of entities for entity-typing via clustering (Section 2.3). They then use this information for disambiguating polysemous predicates (where entities are the arguments). On a WordNet based question-answering dataset, that checks if the question predicate is a hypernym of the candidate answer predicate (using any WordNet sense of either term), they report an accuracy of 94.8% on 211 answers. Results of the entity-typing model are not reported.

Herbelot and Vecchi (2015) show that quantified and specific properties of simple subject/predicate pairs (categories and their properties) can be extracted from standard distributional data (for example, *some cats are black*). Using a supervised least-squares regression setup they report a correlation of 0.66 with the gold standard (equal to the human annotated correlations) on predicting properties of categories. And, a precision of 0.61 on quantifier associations with categories. Their experiments show that predicting properties (that would represent hypernymys and hyponyms in a taxonomy) of categories and their quantifiation can be performed using simple supervised setups with correlations reaching human performance. Predicting fine-grained properties of entities, on the other hand, is not explored (we cover this investigation as an experiment in Chapter 5 and 6).

Kruszewski et al. (2015) map distributional representations of entities and categories to a boolean distributional semantic space by classifying entailing and non-entailing word and sentence pairs. The boolean vectors are an intermediate output of the distributional entailment classifier. The

aim is to to derive a semantic space where dimensions represent sets of entities and categories and the boolean vectors are analogous to the distributional vectors with respect to the distributional inclusion hypothesis (Geffet and Dagan, 2005). They show that performance of a linear classifier using boolean vectors is is at par with a Support Vector Machine classifier using distributional representations.

Herbelot and Baroni (2017) predict distributional meaning representations of new (unobserved at training) entities and categories occurring in known contexts. The contexts here are the one-sentence definitions that these words occur with in Wikipedia. They modify the Word2Vec CBOW model (see Section 2.1.2.2) to selectively train on targets (entities and categories) with their definitions as input. They report a Mean Reciprocal Rank of 0.04 on 300 test datapoints (entity *or* category + one-sentence definition). While they do not analyse entities and categories separately, their results indicate that both are relatively easy to learn given that the definitions (contexts) are informative enough.

To sum up, previous work has used distributional representations of entities and categories (sometimes exclusively) to: 1) aid downstream NLP applications; 2) create novel semantic spaces; and, 3) build more informative representations. There is a clearly established theoretical distinction between the two but, there is no empirically established distributional evidence. We find some focus on the informational analysis of category representations but similar attention has not been given to entities. We, on the other hand, focus on both entities and categories with the purpose of contrasting them in a distributional space.

Herbelot (2015) also investigate distributional representations of entities, just like we do. However, their entity representations are built from categories which instantiate that entity. For example, in their approach, the representation of the entity *Abraham Lincoln* is constructed from its instantiating categories: *man, lawyer, president, leader, etc.*. In a manner of speaking, this is an 'artificial' entity representation. They contrast these

entity representations against the ones constructed from standard distributional methods. Their premise is that the latter do not capture certain necessary properties that are inherent to entities – uniqueness, individuality (i.e., they are different from concepts) and instantiation (i.e., the entity should have a learnable relationship with its instantiating concept). Our work, while being similar in spirit, differs in the following crucial aspects: 1) we analyse the distributional geometry of 'real' entity and category representations built from standard distributional models; and, 2) we show that the instantiation relation between entities and their categories is recoverable by using representations built from corpora. In other words, standard distributional representations of entities capture instantiation.

3.2.2 Instantiation

In this sub-section, we compare instantiation with its closest lexical relation, hypernymy. Moreover, since the modelling of instantiation entails categorization, we also contrast the differences in the formulation of our task and our approach with other categorization tasks like hypernymy detection, Named Entity Recognition and Classification and Named Entity Typing.

3.2.2.1 Versus Hypernymy Detection

Hypernymy is a taxonomical lexical relation that holds a subordinate-superordinate relationship between two common nouns denoting categories (or types). Amongst lexical relations, hypernymy is perhaps the most prominent due to its importance as a key organizational component of semantic memory (Murphy, 2004) and its prominence in formal lexicons, like WordNet (for a detailed definition refer to Section 1.1.1). Due to this, it plays a critical role in many NLP tasks such as lexical entailment (Geffet and Dagan, 2005; Vulić et al., 2017), taxonomy detection (Shwartz et al., 2017), question-answering (Huang et al., 2008) and cross-lingual inference detection (Upadhyay et al., 2018).

While instantiation is a different semantic relation than hypernymy, the two are partially comparable nonetheless. Instantiation, like hypernymy, represents a subordinate-superordinate taxonomical relation between word *types*. However, lexically, the antecedent in an instantiation word-pair denotes an *instance* whereas in hypernymy it denotes a *concept*.

Hypernymy Detection, as a task, is formulated in a manner similar to the detection of other classical lexical relations: given a word-pair, the task is to identify whether that word-pair instantiates that lexical relation or not. There is a comprehensive body of work on distributional modelling of hypernymy (Baroni and Lenci, 2011; Lenci and Benotto, 2012; Roller et al., 2014; Santus et al., 2014; Shwartz et al., 2016). Hypernymy detection can be modelled in both unsupervised and supervised manner. We briefly discuss both approaches in consecutive order in the next two paragraphs.

Unsupervised approaches: Baroni and Lenci (2011) use semantic similarity as a measure to detect hypernymy on their BLESS dataset[3] containing 200 target concepts and report an Average Precision[4] of 0.23. Whereas Lenci and Benotto (2012) use *invCL*, a measure based on the Distributional Inclusion Hypothesis[5], that computes a score over the weighted feature inclusion and non-inclusion of a word-pair. They report an improved Average Precision of 0.40 on the BLESS dataset. Santus et al. (2014) detect hypernymy as a measure of the product between: 1) the reciprocal difference of the semantic generality of the word-pair; and, 2) the vector cosines of the word-pair. The semantic generality of a word is the entropy of its most associated contexts identified through Local Mutual Information

[3]Baroni and Lenci (2011) designed this benchmark dataset specifically to evaluate distributional semantic models on various semantic relations. The dataset consists of tuples, each denoting a semantic relation between two concepts, in the form of (target concept, semantic relation, relatum concept). The dataset has 200 target concepts (that are single-word nouns in the singular form) divided in 17 broad classes like, *building, tool, tree and weapon*. It covers 5 relations: *co-hyponymy, hypernymy, meronymy, attribute and event*. The relatum are selected from semantic (WordNet) and text (Wikipedia) sources.

[4]A method derived from Information Retrieval and combining precision, relevance ranking and overall recall (Lenci and Benotto, 2012).

[5]The Distributional Inclusion Hypothesis states that given a hypernym W and its distributional contexts C, the contexts of its hyponyms i.e. $\hat{C}$, occur in the subset of C (Geffet and Dagan, 2005; Zhitomirsky-Geffet and Dagan, 2009).

(Evert, 2005). They report an AP of 0.59 on a subset of the BLESS dataset containing 1,277 hypernymy-related word-pairs.

Supervised Approaches: An effective approach for hypernymy detection is to model the task as a supervised binary classification problem, where the word-pairs are given to the classifier as input and the output is a binary prediction: 1 (or, *yes*; if hypernymy holds) and 0 (or *no*; if the input does not reflect hypernymy). In other words, modelling hypernymy detection is, in essence, a discriminative task of learning and predicting between two classes, *positive* and *negative*, where the model learns to discriminate through an artificially induced dataset that reflects the instantiation of the semantic relation between two words through positive and negative examples. The aim is to see if a model can learn to generalize the instantiation of a relation via the lexical patterns captured by meaning representations.

Baroni et al. (2012) model hypernymy through a Support Vector Machine (SVM) classifier and report an accuracy of 88.6% on their ENTAILMENT balanced dataset (of 1385 positive + 1385 negative examples) created from the WordNet noun hierarchy. In addition to the SVM, Roller et al. (2014) also use a Logistic Regression classifier on the BLESS and ENTAILMENT hypernymy dataset. They report an AP of 0.84 and 0.85, respectively. Shwartz et al. (2016) first compute a representation of the dependency paths of a hypernymy word-pair (extracted from a parsed corpus) using a Long Short Term Memory (LSTM). This path vector is then concatenated with the distributional vectors of the word-pair and given as input to a feed-forward network classifier. They create a dataset of 28,295 hypernymy word-pairs which prohibits memorization through lexical overlap (extracted from WordNet (Fellbaum, 1998), DBPedia (Auer et al., 2007), WikiData (Vrandečić and Krötzsch, 2014) and Yago (Suchanek et al., 2007)) on which they report an F_1-score of 0.70. Note that they choose to not use the BLESS and ENTAILMENT datasets owing to their relatively small size.

Within the supervised setups, there are a few ways to represent the distributional features (of word-pairs) given at input: concatenation $(\vec{x} \oplus \vec{y})$, difference $(\vec{x} - \vec{y})$, Asym $(\vec{x} - \vec{y})^2$, dot product $(\vec{x} \odot \vec{y})$ and component-wise addition $(\vec{x} + \vec{y})$. The first two, *concatenation* and *difference* reportedly give the best performance (Baroni et al., 2012; Roller et al., 2014) due to the former being the most informative input (feature-wise) whilst the latter being the best at capturing distributional inclusion.

While the unsupervised approaches are advantageous due to the fact that they do not require labelled data, the supervised approaches the clear winners in this task. One main reason is that these approaches allow a model to learn from both positive and negative examples of a task which increases the ability of the model to generalize better over unseen data. We model *instantiation detection* (as described in Section 3.5.1) based on the most effective modelling approaches for *hypernymy detection*. We design it as a supervised binary classification problem and experiment with all the above mentioned input representations but, report results and analysis on those which give the best accuracies.

3.2.2.2 Versus Named Entity Recognition and Classification (NERC)

As we describe in Section 2.5, NERC is the task of recognizing the occurrence of Named Entities (NEs) in unstructured text and subsequently categorizing the identified NEs to a type. Entity recognition is viewed as a sequence labelling problem at the token level and the subsequent categorization of the identified NE is seen as a multi-class classification problem[6]. Traditionally, NERC was implemented using NE language resources, like Gazetteers, and hand-crafted rules consisting of lexical patterns that denote entity occurrences (Krupka, 1995). In statistical modelling, Conditional

[6]In a multi-class classification, an input is classified into one class out of many classes. This is different from multi-label classification where an input can be categorized into more than one class based on the assumption that the membership of that input is partially shared by many classes. Binary classification, on the other hand, implies that an input can be classified into one of two classes.

Random Fields (CRFs) and sequential learners (like Recurrent Neural Network (RNN) and its variants – Long Short Term Memory (LSTM) and Bi-directional LSTMs) are the popular choices for modelling entity recognition and categorization. We discuss some examples as an elaboration on the above.

Lin and Wu (2009) train a CRF over the CoNLL 2003 Shared Task dataset (Tjong Kim Sang and De Meulder, 2003) that consists of: 203,621 tokens for training, 46,435 tokens for testing and entities of four types – persons (PER), organizations (ORG), locations (LOC) and miscellaneous names (MISC)[7]. They also cluster distributional representations of 20 million (predominantly noun) phrases into 3000 clusters. If the tokens of an input sequence match the tokens of a particular phrase, they optimize the CRF with additional features of the cluster the phrase belongs to. The cluster features were found to boost model confidence during classification when compared to inducing the CRF with additional syntactic features (like, part-of-speech labels, etc). The evaluation results in an F_1-score of 90.90; improving about 7.12 points over the no-clustering CRF baseline of 83.78.

Lample et al. (2016) use a Bi-directional LSTM to model character-level word representations and concatenate them with pretrained token-level word embeddings in order to capture both orthographic and distributional sensitivity (respectively) of a token sequence containing named entities. Orthographic features allow a model to learn what a word denoting a named entity looks like (at the level of occurrence) in a token sequence. Distributional features allow the model to learn the position of named entities through contextual information. The output token sequence is then given as input to a CRF classifier for type categorization. On the CoNLL 2003 dataset, they report an F_1-score of 90.94. This is not a significant increase over Lin and Wu (2009) but, the novelty in this lack of use of

[7] The CoNLL NERC datasets are the most popular datasets for testing novel state-of-the-art modelling approaches as they provide time-tested baselines that the new models can evaluate against. The one shortcoming of the CoNLL NERC datasets is that the number of entity types is limited to a flat hierarchy of about 10-20 types (see Section 2.5.1 for more details).

externally labelled data when compared to previous work. In contrast, Ma and Hovy (2016) use a Convolutional Neural Network (CNN) instead of an LSTM to model character-level information. On the same dataset they report an F_1-score of 91.21.

Compared to our task of instantiation detection, where we focus on the modelling of the lexical relation of instantiation over an already available entity collection, the focus in NERC is on entity recognition through sequence labelling *per se* and subsequent entity classification. In fact, the space we use is built on top of a corpus processed with an NERC system. Moreover, in traditional NERC systems the categories are rather coarse-grained (e.g., location, person, organization, other). And although there is a tendency towards more fine-grained named entity classification (see Ling and Weld, 2012a, and the next section), we classify on a much larger set of fine-grained categories.

3.2.2.3 Versus Named Entity Typing (NET)

NET (or entity-typing) is the task of detecting the type(s) of a Named Entity (NE) in context (Del Corro et al., 2015). In other words: given a sentence (or more generally, a linguistic expression) in which an NE has already been identified, that NE has to be categorized in one or more types depending upon its neighbouring words[8]. Consequently, the task is viewed as a multi-label classification problem.

NET, like NERC, is modelled at the level of token sequences. However, both are complimentary because: 1) in NET, NE tagging is assumed; and, 2) NET is performed on fine-grained types (when compared with NERC) usually extracted from large-scale taxonomical resources, like knowledge-bases and WordNet. In essence, NERC can be followed up by NET; where the coarse types identified during NERC can be used as features, or even

[8]For example, in the sentence “Barack Obama is the first black president of the United States of America", *Barack Obama* and *United States of America* are the identified NEs. Consequently, the goal of entity-typing is to infer that *Barack Obama* belongs to the types (*president, politician, african-american, person*) and *United States of America* belongs to the type *country*.

simply as hypernyms, to obtain a much finer and accurate level of NE categorization. We list out a few, but varied, modelling approaches for entity-typing.

Del Corro et al. (2015) build a 16K type resource by extracting type information from the WordNet noun hierarchy and Freebase as well as by applying pattern based matching, nominalization (transformation of verb into deverbal nouns) and corpus based semantic similarities of NE from noun phrases. They train a Naive Bayes classifier to select the most appropriate type given its context. On three datasets: 500 randomly selected sentences from both, New York Times corpus (Sandhaus, 2008) and CoNLL 2003 NERC shared task (introduced in Section 3.2.2.2) and, 100 NE tweets from Twitter; they report a precision of 72.42, 80.66 and 66.35 respectively on fine-grained types. Note that this modelling approach can be used on any domain without requiring full supervision. Consequently, this requires the infusion of additional syntactic and semantic features which are extracted from other external tools.

Ma et al. (2016) employ a supervised approach for entity-typing. Their entity representations are constructed from text using the Skip-gram model (see Section 2.1.2.2). The type representations are based on a prototype-driven approach where for each type, a prototype set of entities is computed and their representations are averaged. The prototypes are the top k NE mentions (where k=60) computed by Normalized Point-wise Mutual Information (NPMI – see footnote on Page 22) between the type and the NEs. Based on the model by Yogatama et al. (2015), the entities and types are projected into a joint space such that each entity is close to its type. The model is a bi-linear scoring function with a ranking based loss which encourages the model to place positive types above negative types. They report a Micro-F_1-score of 76.50, 59.08, 66.53 on three datasets: BBN dataset (Weischedel and Brunstein, 2005), OntoNotes dataset (Weischedel et al., n.d.) and Wikipedia dataset (Ling and Weld, 2012a) having 47, 89 and 113 unique types respectively. Although the number of types being modelled is relatively small, this model uses less parameters as compared to

linear and non-linear models. Thus, make it computationally cost- effective.

Choi et al. (2018), on the other hand, define entity-typing as the task of predicting a set of natural language phrases that describe the type(s) of an NE, given a NE in context. In their modelling approach, inspired from the AttentiveNER model by Shimaoka et al. (2017), an attention based LSTM is used to learn a contextualized representation of the token sequence (excluding the NE) which is concatenated with the NE representation(s) to predict type representation(s). The NE representations are the concatenation of two items: 1) a character-based representation produced by a CNN on the NE; and, 2) a weighted sum of the pre-trained word representations of the NE. The type representations are learnt through a combination of the weighted features of general, fine, and ultra-fine types. They test their model on a novel crowd-sourced dataset, containing 6000 examples with over 2,500 unique types, and report an F_1-score of 32.0.

The above work in NET (and the references therein) show that in NET, the focus is on categorizing NEs into increasingly finer types. The scores are found to suffer due to a generally low recall which is attributed to the categorization of NEs into relatively high number of fine-grained types. Another major challenge is that the task, by its definition, requires the types to be inferred from the contexts. And, it is entirely possible that for a specific type (that an NE belongs to) the contextual information might not be explicit enough; in the above example, *Barack Obama* is an *african-american* might seem cognitively straight-forward to a human reader but, it is a difficult type to infer distributionally in the given context. The modelling approaches over time are seen to increase in complexity as the number of types grow and there is substantial effort towards optimizing the contextualized representations of NEs and types. Having said that, it is surprising that there is little information (and analyses) in NET literature on the NEs and types that are difficult to learn and the criteria under which categorization fails.

With both NERC and NET in perspective, entity categorization can be viewed as a limited form of our instantiation detection task. This is because

we analyze the entity representations themselves and tackle a wider set of tasks related to instantiation and in general, lexical relational modelling. Nonetheless, our focus on public entities is shared with both NERC and NET. However, we ask our models *not* to disambiguate corpus occurrences of named entities in context, but to assess pairs of entities and categories without context just on the basis of representations for the entities. We believe that this tests the ability of our models to acquire the range of possible categories for the entities.

3.3 Instantiation Dataset

Since there is no previous dataset on instantiation, our first contribution is the creation of a benchmark dataset for this relation with positive and negative examples (word pairs of entities and categories). For entities, as motivated in Section 1.3, we work with representations of Named Entities (NEs) in the public sphere such as *Italy* or *George Washington*, as opposed to private named entities like *my neighbor "Michael Smith"* or unnamed entities like *"the bird" I saw today*. We also do not take into account other referential expressions (*he*, *the 1st president of the United States* or *the president*.

Our preference towards NEs is because it is possible to obtain distributed representations for those using standard methods in distributional semantics and deep learning. This is common to related work on entities in distributional semantics. In computational linguistics it is general practice to use state-of-the-art pretrained meaning representations for domain independent tasks (Sienčnik, 2015; Şulea et al., 2016; Das et al., 2019). The same is also true for Information Extraction, Knowledge Representation and Semantic Web communities which are not primarily concerned with the meaning construction frameworks but rather using the state-of-the-art meaning representations for NLP applications, like question-answering (Khot et al., 2018), relation prediction (Bordes et al., 2013) and knowledge

base population (Basile et al., 2016) amongst others; although some tasks may further optimize these representations for higher accuracies.

We follow the procedures of previous studies that created benchmark datasets for lexical relations (Baroni and Lenci, 2011; Roller et al., 2014; Levy et al., 2015b). The datapoints in these datasets consist of *source–target* (binary) word-pairs which express the semantic relation they are supposed to instantiate, called *positive examples*. A source word can have multiple targets resulting in an individual datapoint for each source-target combination. With the intention to induce generalized learning (so that the machine learning models to not memorize patterns), the datapoints also consist of confounders, also called *negative examples*, where each source word is paired with a random target word that does not express the semantic relation the source is originally associated with in the positive set of datapoints. With that in perspective, we extract the positive *entity – category* examples, such as (*Virginia Woolf – writer*), from WordNet. We pair them with confounders, like (*Virginia Woolf – athlete*), in a principled way, partially inspired by Baroni and Lenci (2011). Our aim is to enable a structured investigation of the different aspects involved in instantiation, in particular the distinction between entity and category in general vs. instantiation proper and the role of similarity.

3.3.1 The Distributional Space

The entity and category meaning representations, that we use for our experiments and analyses, are from the distributional space[9] made available by Mikolov et al. (2013b). The space has been constructed from the Word2Vec toolkit and uses the Skip-gram architecture (see Section 2.1.2.2) to construct 1000-dimensional pretrained vectors from the 100B word Google News corpus (primarily consisting of news articles). In all, the space consists of a collection of 1.4 million vectors whose targets are drawn from Freebase and, are labelled according to their Freebase identifiers. The

[9]The space can be downloaded from the following link: `https://code.google.com/archive/p/word2vec/`

identifiers follow the Freebase naming convention to label any object as a single lower-cased token prefixed with a (now deprecated) '*/en/*'. In case of multi-word expressions, as commonly observed in the case of named entities, the lower-cased words are concatenated with an underscore (_) to form a single token. For example, the vector for the entity *George Washington* is labelled as */en/george_washington* in the space.

The collection includes both named entities (denoted by proper nouns) as well as categories (denoted by common nouns). And, to the best of our knowledge, during the implementation of this work, this space was the largest resource for entity vectors. Another reason to use this space is that, in conformance to the definition of our data selection strategy (above), for each entity, the space also contains its instantiating category.

We rescale the vector values column-wise so they lie within the [-1, 1] range. We treat the vectors as static, rather than optimizing them for the task at hand, because we are interested in the geometric structures present in a generic word meaning space, following the rationale of the Distributional Memory (Baroni and Lenci, 2010). Moreover, optimizing the vectors might overfit on the instantiation relation on our dataset.

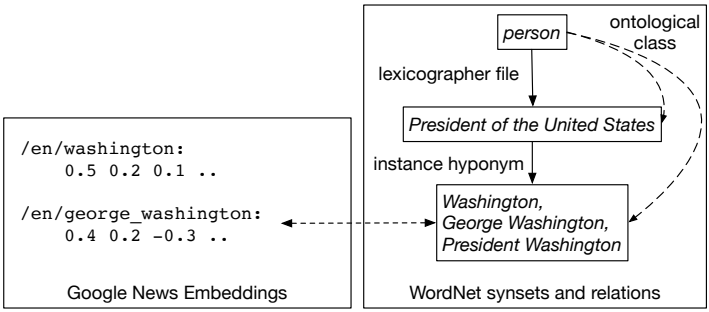
The next two subsections discuss the positive datapoints and the confounders, respectively.

3.3.2 Positive Datapoints

Our method for obtaining positive datapoints is shown in Figure 3.1. We start out with the WordNet noun hierarchy and extract all pairs of WordNet synsets (e, c) that are linked with the *instance_hypernym* relation. This ensures that each e is an entity and c a category.

Next, we retain only those pairs for which we have coverage in our distributional space (Section 3.3.1). Mapping between WordNet synsets and Freebase identifiers (from the distributional space) requires some heuristics. Notably, WordNet synsets with multiple elements can map onto several

FIGURE 3.1: Mapping between WordNet synsets and Google News targets.



Freebase identifiers. For example, the synset *(Washington, George Washington, President Washington)* maps onto two Freebase identifiers, and thus, two representations: *Washington* and *George Washington*[10]. We choose the Freebase identifier that matches the longest token sequence of the WordNet synset, as it will be the least ambiguous; in the sense that while *Washington* can refer to different entities (a person, a state, or a city), *George Washington* on the other hand almost always refers to the former president of the USA. Of course, the representation for *George Washington* will be built from fewer occurrences as compared to when all words (aliases) denoting the person *George Washington* are taken into account but, our dataset design favours precision over recall. Note that this strategy reduces, but does not completely remove the incidence of short Freebase identifiers.

The resulting dataset is a collection of 5,469 positive cases of instantiation, with 4,750 unique entities and 577 unique categories (see Table 3.1). There are more positive word pairs than entities because some entities in WordNet belong to more than one category: For example, (*George Washington – president of the United States*), (*George Washington – general*), since *George Washington* was both a president and a general. There is also

[10]Note that as mentioned in Section 3.3.1, multi-word named entities are included in the distributional space as single tokens (but not expressions like *President Washington*) (see Mikolov et al., 2013b, for details on how they identified multi-word expressions).

TABLE 3.1: Positive datapoints: Statistics and examples by ontological class.

Ontological Class	#Datapoints	#Entities	#Categories	Example
person	2742	2330	294	Madame Curie – chemist
location	1746	1512	99	Oaxaca – city
object	633	630	59	Nile – river
artifact	121	118	55	Bastille – fortress
communication	98	97	43	Iliad – epos
act	69	67	20	Alamo – siege
other	60	60	26	Paleocene – epoch
Total unique	5469	4750	577	–

some remaining ambiguity, in the form of different entities with the same name that belong to different categories – e.g. *William Gilbert the poet* vs. *the physicist*. There are no automatic means to distinguish if the vector for *William Gilbert* represents the poet or the physicist and since the two examples result from natural properties of instantiation, we use the same vector for both and perform no further filtering.

Note that there are many more entities than categories, which is to be expected because there are more entities than (lexicalized) categories, and some categories contain a large number of entities. The effect may however be reinforced by the way the Google News space was built: its vocabulary consists of the nodes in Freebase, a database primarily geared towards entities, though it also had nodes for categories.

For analysis (in Section 3.4) and negative datapoint selection, we additionally use the WordNet LEXICOGRAPHER FILE labels, which can be used as proxies for semantic or ontological classes [11] (Rigau et al., 1997; Curran, 2005). These are shown as rows in Table 3.1 (also see Figure 3.1). Most of the datapoints belong to ontological classes *person* and *location*. The *person* class consists of popular and well known, fictional and non-fictional, historical as well as modern day people; *location* contains geopolitical

[11] If the entity and category are assigned different lexicographer files, we use the one of the entity; if one of them is missing a class, we use the other one. This affects a total of 329 (6%) datapoints. Also, we collapse all ontological classes with fewer than 50 occurrences in our data into a class *other*.

TABLE 3.2: Examples of confounders. POTUS = President of the United States.

Type	Example 1	Example 2
Positive	George Washington – POTUS	Mumbai – city
Inverse	POTUS – George Washington	city – Mumbai
Inst2Inst	George Washington – Peter Behrens	Mumbai – Vicksburg
NotInst-global	George Washington – river	Mumbai – statesman
NotInst-inClass	George Washington – astronomer	Mumbai – residential area

entities, like countries or cities; *object* mostly consists of geographical and natural entities; *communication* includes literary texts but also computer programs and operating systems; *artifact* covers all kinds of man-made entities; finally, *act* consists of famous events.

3.3.3 Confounders

As mentioned above, we include different types of confounders in our dataset. In our experiments in instantiation detection (Sections 3.5 below, Section 4.1.4 and Section 4.2.1 in Chapter 4), we ask the models to distinguish between positive examples and confounders. More specifically, we generate four sets of confounders by transforming each entity-category positive example (e, c) as follows:

Inverse Swap the positions of entity and category, yielding (c, e).

Inst2Inst Replace the correct category by a different random entity e' of the same ontological class, yielding (e, e').

NotInst-global Replace the correct category c by a random wrong category c'', from the global distribution of categories, yielding (e, c'').

NotInst-inClass Replace the correct category c by a wrong category c', this time sampling from the same ontological class, yielding (e, c').

Inverse tests that the models correctly capture the asymmetric nature of instantiation. *Inst2Inst* checks that the models are not fooled by similarity

(entities in the same ontological class are similar to each other, see Section 3.4). Finally, *NotInst-global* and *NotInst-inClass* aim at testing that models actually learn the relation between a specific entity and a specific category, as opposed to learning to classify entities vs. categories in general (Levy et al., 2015b). The difference between *NotInst-inClass* and *NotInst-global* is one of difficulty: in *NotInst-inClass*, confounder categories come from the same ontological class as the correct categories, and thus are semantically more similar to the correct category (e.g. pairing *George Washington* with *astronomer*) than in *NotInst-global* (where *George Washington* is paired with *river*). Table 3.2 shows two examples of a positive datapoint with its corresponding confounders.

When pairing confounders with the positive examples, we obtain in four different balanced subsets, consisting of pairs of expressions for which the instantiation relationship either holds (positive examples) or does not (confounders): Pos+*Inverse*, Pos+*Inst2Inst*, Pos+*NotInst-global* and Pos+*NotInst-inClass*. Two final variants, Pos+*Union-inClass* and Pos+*Union-global*, combine the positive examples with *Inverse*, *Inst2Inst*, and one of the two *NotInst* variants, respectively. These two *union variants* are more challenging in that they require models to distinguish positive examples from confounders of different types, and have a 1:3 positive-to-negative ratio.

3.3.4 Dataset Partitioning and Memorization

We split each dataset variant into training, validation and test sets (80, 10, and 10% respectively). This is however not enough to make sure that the models can generalize: the related task of hypernymy detection has been shown to suffer from the problem of *memorization* (Roller et al., 2014; Levy et al., 2015b), that is, models learning by heart that certain words (such as *animal*) make good hypernyms instead of truly learning the hypernymy relation. The problem is that, in a naïve random split of the datapoints, even if the pairs are not reused across partitions, individual members of

the pair can be. Thus, good results can hide a lack of generalization, in particular for frequent categories.

To address this issue, we adopt a variation of the methodology of Roller et al. (2014) which ensures that there is zero lexical overlap between training, validation, and test sets. Specifically, we split the test set into many equal-sized test folds and remove overlap with the training and validation data: For example, if *(George Washington, President of the United States)* occurs in a test fold, then all pairs containing either *George Washington* or *President of the United States* are removed from the corresponding training and validation data. We choose the number of test folds such that the average size of the training set after removing the lexical overlap is 90% of the original training data (fewer, and therefore larger, test folds lead to more excluded training data). This results in 83 test folds.

The *Leave-one-out* evaluation, as chosen by Roller and Erk (2016), would have increased computational load substantially despite being a better approach to remove lexical overlap, in theory. That is because in this strategy each test fold contains datapoints that have exactly one unique antecedent. This unique antecedent is selected from the set of antecedents that are pooled from the entire list of word-pairs in the dataset. For instance, in a hypothetical three word-pair dataset of (*George Washington – president*), (*George Washington – politician*) and (*Abraham Lincoln – lawyer*), *George Washington* and *Abraham Lincoln* are the unique antecedents resulting in two test folds: the first containing the two former datapoints and the second containing the third datapoint. The remaining datapoints, not included in the test fold, constitute the training data. While this strategy too ensures that there is zero lexical overlap, it also results in as many cross folds over the data as is the size of the set of antecedents. For example, even on our moderately sized dataset, with 5469 word-pairs having 4750 unique entities, the number of cross folds would be 4750. On large scale datasets, where the number of unique antecedents can run into thousands (or more), this strategy can quickly become computationally infeasible.

3.4 Data Analysis: Entities and Categories in Space

In Section 3.1, we put forward our hypothesis that entities and categories are represented distinctly in a distributional space based on the observation that this distinction is also acknowledged ontologically as well as in formal semantics.

In this section, we perform an exploratory data analysis of entities and categories that exist within our distributional space to validate our hypothesis. A validation holds importance because its affirmation not only provides support to our dataset design decisions in Section 3.3 but also lays the ground-work for making a distinction between instantiation and hypernymy in Chapter 4.

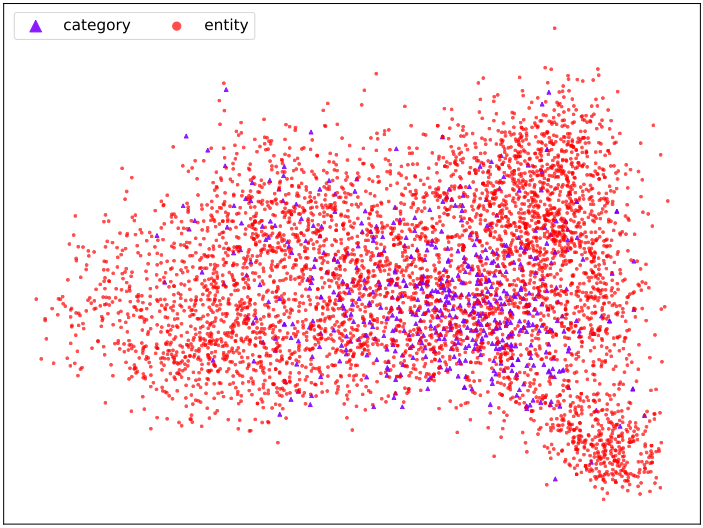
We carry out this analysis on the positive portion of our dataset (Section 3.3.1) and discuss our findings with respect to the distributional geometry of entities and categories.

3.4.1 Layout of Entities and Categories in Space

As a preliminary step, we conduct a visual inspection of the entities and categories in the distributional space. We expect that the categories, being informationally generic, would tend to have a certain level of semantic similarity with other categories. Thus, their vectors would congregate towards the center of the space as well as live close to each other. On the other hand, the entities being informationally specific, would be distributed all over the space. However, given that the entities are instances of categories in the same space, we also expect entities to cluster towards their instantiating categories.

Figures 3.2 and 3.3 represent the first two dimensions of a Principal Component Analysis (PCA) transformation of the original 1000-dimensional vectors. Note that PCA transformation creates a new space by looking at

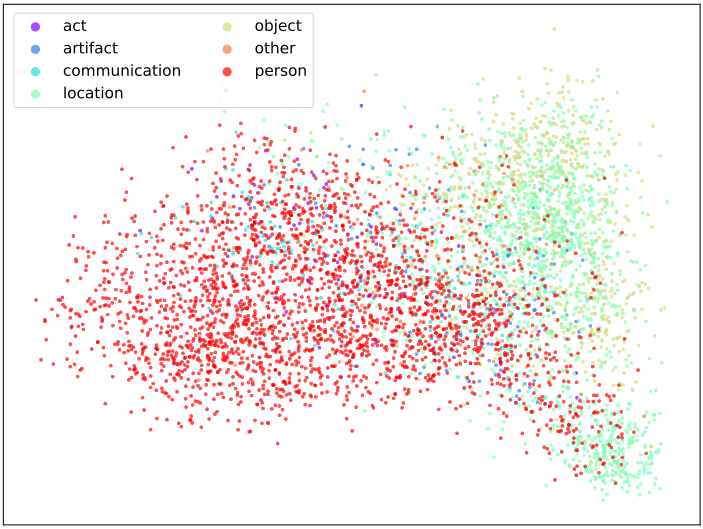
FIGURE 3.2: Entities and categories in distributional space, first two PCA dimensions.



properties that show as much variation across classes as possible (see footnote on page 35). Consequently, the resultant low-dimensional vectors are (at best) an approximation of their high-dimensional counterparts. Nonetheless, PCA is an effective technique to generate (2D and 3D) visualizations for an exploratory analysis of large amounts of high-dimensional data in a fast and computationally feasible manner.

Figure 3.2 shows that entities and categories are not in distinct regions of this reduced space, hence, the two (in the entirety of the data) are not linearly separable. Categories seem to be concentrated towards the center, forming a roughly defined radial structure. On the other hand, entities are much widely dispersed in the graph. This fits our expectations. Since the entities and categories are intermingled (visible in-and-around the center), this serves as an indication that the distributional features that bring about these distinctions might be encoded in higher dimensions and their exploration requires more than a visual inspection (we explore this further

FIGURE 3.3: Ontological classes



in Section 3.4.2). However, contrary to our expectation – that entities would live close to their instantiating categories, there are regions in the space that are occupied only by entities and not categories (see bottom- left, bottom-right and top-right in the figure). In other words, some categories are more similar to other categories, i.e. they are closer in meaning to each other, as compared to their instantiated entities. And, while this increases the challenge of modelling instantiation, we take this as direct evidence that at least a large chunk of entities are quite distinct from their instantiating categories. Thus, adding to the validation of our hypothesis that entities and categories are distributionally distinct.

Figure 3.3 shows that the data is also sensitive to ontological distinctions such as those encoded in WordNet (see Table 3.1, Section 3.3). The figure shows a clear division between animate (*person*, left) and inanimate entities/categories (*location* and *object*, right; although these two classes largely overlap, *object* is more concentrated in the upper right part and *location* in the lower right part). In the middle, partially overlapping with

the classes above, we find the smaller classes *artifact*, *act* (human-centered events), and *communication*, as well as the catch-all *other* class.

3.4.2 Clustering Analysis of Entities and Categories Representations

From the observations of Figure 3.2 and 3.3, it is clear that distinctions between entities and categories exist but are not entirely manifested in reduced dimensionality and that there is a broad-level ontological organization seen in the distributional space.

Therefore, we carry out a clustering analysis to explore in-depth if there are latent properties (not apparent in our low-dimensional inspections) that further govern the distributional geometry of entities and categories in the space. As discussed in Section 2.3, clustering is a commonly used exploratory technique that categorizes objects into meaningful groupings without the use of any pre-classified (training) data i.e., in an unsupervised manner. It is highly useful for preliminary assessments about the relationships between data with minimal assumptions and little prior information.

3.4.2.1 Model

We use the standard *K-means* algorithm (see Section 2.3.2 for details) for clustering our data. The algorithm generalizes well to clusters of different shapes and sizes and is known to converge quickly over large datasets. Note that, we do not consider algorithms which create dynamic clusters, like *DBSCAN* (Ali et al., 2010) because these algorithms work best with data which may result in evenly distributed clusters, which our pre-experiment analysis and data do not indicate.

We build the clustering model from the *SciKitLearn* package (Pedregosa et al., 2011) with clustering solutions ranging from 2 to 15. Our range of clustering solutions is determined by the want to identify higher-level distinctions between entities and categories with respect to certain criteria like, organization by types or categories or ontology or topicality. As a

design decision, we deliberately restrict to an upper limit of a clustering solution with 15 clusters. Since the categories are somewhat centrally organized and entities are intermingled with them, a larger number of clusters would not contribute towards addressing our current hypothesis. On the contrary, an increasingly higher number of clusters would incrementally bring forward finer properties shared by entities and categories; which is not our focus.

Hyperparameters and evaluation metric. Since the centroid seeds are randomly initialized, the algorithm is run 10 times with different seeds and the model with the best performance in terms of inertia is selected. Inertia is a within- cluster sum-of-squares criterion that is computed with each iteration and calculated to be within a predefined threshold – *tolerance*, to indicate model convergence. It is akin to the *loss/cost* function in a neural network. The number of iterations for each model run is set to a maximum of 10000 with convergence at tolerance value of 1.0 [12]. Since 'inertia' is not a normalized metric, tolerance is not defined between [0,1]; although lower values are better and 0 is considered optimal.

The evaluation criterion used is Adjusted Mutual Information (AMI) which computes the score of mutual information shared between the gold cluster indices and the predicted cluster indices, adjusted to account for chance. The metric results in a score of 1.0 when two clusterings are identical and around 0.0 when the predicted clusterings are similar to random partitions of data.

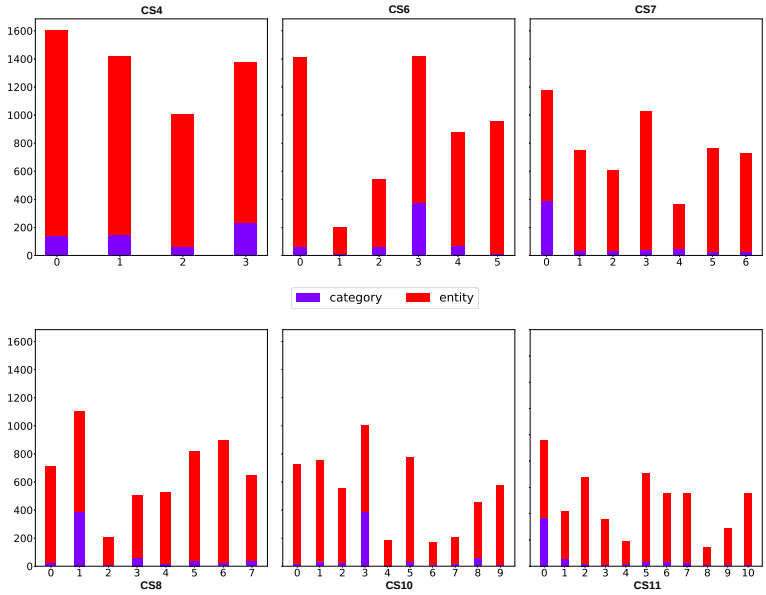
3.4.2.2 Results and Discussion

We run the clustering model on the 1000 dimensional entity and category vectors in the positive dataset and analyse the results against the entity/category axis as well as WordNet ontological classes.

Figure 3.4 shows a representative selection of the top-6 clustering solutions (according to AMI) along with the distribution of entities and

[12] We varied the tolerance between $[10^{+1}, 10^{-6} \mid step\ ratio = 10^{-1}]$ and found 1.0 to be optimal.

FIGURE 3.4: Distribution of entities vs. categories for top AMI-scoring clustering solutions.



categories for each of the clusters within a solution. They range from 4 clusters at the top left to 11 clusters at the bottom right. The figure shows a similar trend to Figure 3.2: On the one hand, the clustering algorithm does not use the division between entities and categories as its primary organizing principle for cluster solutions, especially not with a small numbers of clusters (we will see below that it uses the animate/inanimate division); on the other, for solutions with a higher number of clusters, categories tend to group together in a single cluster (cluster 3 in clustering solution 6 (or CS6, in short) as well as CS10, cluster 0 in CS7 and CS11, cluster 1 in CS8). This is consistently found in all solutions with higher number of clusters (not shown). This *conceptual cluster* is mainly a concentration of *person* categories which reflect professional/societal roles, like *musician, physicist, minister, king, environmentalist, engineer, artist, etc*. Note that this cluster also contains many entities; they are instances of these categories.

FIGURE 3.5: Distribution of WordNet ontological classes for top AMI-scoring clustering solutions.

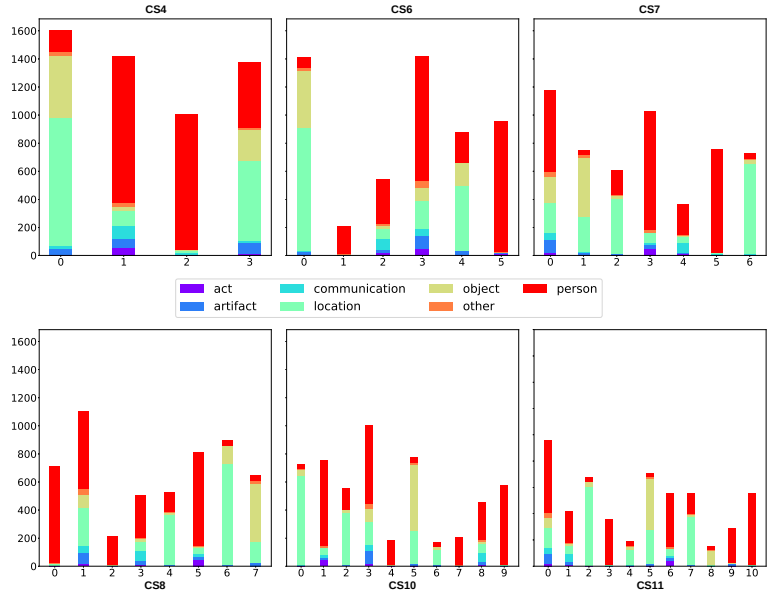


Figure 3.5 presents the same 6 clustering solutions, this time showing the distribution of the WordNet ontological classes. As could be expected, the animate/inanimate distinction appears prominently: Clusters 0 and 3 in CS4 contain most *object* and *location* datapoints, whereas clusters 1 and 2 contain most *person* datapoints. However, while the *person*/*location*/*object* distinction is relevant throughout the clustering solutions, we do not see a clear mapping between clusters and ontological classes in general. Therefore, we carry out a more detailed manual inspection of the clustering solutions to spot any other regularities that might emerge out of the clusters. What becomes evident is that the current cluster compositions are based on topical distinctions and with the increase in the number of clusters, the clusters tend to get more topic specific.

In Table 3.3, we list our observations on topical regularities from three domains (*person*, *location*, *object*) on three clustering solutions (CS7,

CS8 and CS11) [13] from the top-6 clustering solutions. We see that there are certain clusters (containing mainly entities) that consistently show strong cohesion across different solutions. For example, 1) the clusters dominated by the *location* class for *Eurasia* and *America*; and, 2) the *Historical and Religious*, the *Writers* and the *Political, Academic and Science* entity clusters of the *person* class. Also, note the grouping of *Astronomers* (*person*) and *Astronomy* (*object*) and *painters* (*person*) in CS8 and CS11. Most of these 'topics' are a congregation of entities by topicality which indicates that the entity representations capture the prototypical representations of topics. This is further supported by the gradual break-down of the *Musicians, Composers and Painters* (*person*) cluster in CS7 to their own individual clusters in CS11. As well as the hierarchical bifurcation of the *Geographical* (*object*) cluster into clusters that represent *larger* and *smaller* geographical sub-categories like, (seas and countries) and (rivers and parks) respectively.

Interestingly, the clustering of entities by topicality in Table 3.3 brings to light the possibility of entity representations being used to construct prototypical representations of categories. However, it remains to be evaluated *how* this prototypical representation compares to the concept-based representation constructed from a corpus. We explore this question further in Chapter 4.

To sum up, both the PCA reduction and the clustering analysis suggest the primacy of ontological and topical dimensions in the distributional space, as far as entities are concerned. Categories, on the other hand, are centrally congregated in Figure 3.2 and they are partitioned into one cluster which is consistent throughout all our clustering solutions in Figure 3.4. That is to say that categories are immune to the broad distinctions we see in entities. This validates our hypothesis that entities and categories are different distributionally.

[13] Our choice is a random selection of solutions with the intent to show topical distributions over an incremental increase in the number of clusters

TABLE 3.3: Prominent topics in representative clustering solutions (from manual analysis).

Clustering	*person*	*location*	*object*
CS7	0: Political, Academics, Science and other professions 3: Historical and Religious 4: Writers 5: Musicians, Composers and Painters	2: Eurasia 6: America	1: Geographical
CS8	0: Musicians and Composers 1: Political, Academics, Science and other professions 2: Painters 3: Writers 5: Historical and Religious 6: Astronomers	4: Eurasia 6: America	6: Astronomy 7: Geographical
CS11	0: Political, Academics and Science 1: Writers 3: Composers 4: Astronomers 6: Historical and Religious 7: Explorers 9: Painters 10: Musicians, other professions	2: America 7: Eurasia	4: Astronomy 5: Larger Geography - (Seas, Oceans, Islands, Continents, Countries) 8: Small Geography - (Rivers, Canals, Parks)

Next, with the validation of our hypothesis above (and consequently, with greater confidence in our dataset definition), we model the lexical relation – instantiation.

3.5 Experiment 1: Instantiation Detection as Classification

As described in Section 3.2.2.1, we frame instantiation detection as a supervised binary classification task based on the most effective approach for modelling a taxonomical lexical relation (Baroni and Lenci, 2011; Roller et al., 2014). In the following sub-sections we describe the experimental setup, the baselines and the evaluation scheme we use for modelling this

task. This is followed by the results of the model and a qualitative analysis on the erroneous predictions. We also discuss the effect of different input representations and feature memorization on the model performance.

3.5.1 Models

We test two generic model architectures. The first one is a simple logistic regression (LR) classifier. The second one is a feed-forward neural network (NN) classifier. The NN classifiers are built with the Keras toolkit (Chollet et al., 2015) with the Theano (Bergstra et al., 2010) back-end.

We choose LR and NN for their comparability: The LR classifier has no hidden layers, thus treating all features as independent, while the NN model introduces hidden layers that can model non-linear relationships between input and output and facilitates interactive behavior between the input features. This setup allows us to gauge to what extent these aspects affect instantiation while staying within the same general modelling framework. For the NN classifier, we experimented with 1-4 hidden layers but found a decline in model performance from 3 layers onwards. Hence, we report results only on the architectures with one hidden layer (NN-1HL) and two hidden layers (NN-2HL).

The input and output of both classifiers are identical: for each datapoint (i.e., for an *entity – category* pair (e, c)), the input is a function of the two vectors, $f(\vec{e}, \vec{c})$ and the output is a binary value indicating whether the entity e is instantiated by the category c. Note that the function f can be any operation (not necessary mathematical) on the vectors $\vec{e}$ and $\vec{c}$. However, for the main experiment we start by considering the most straightforward input (but also proven to be effective (Baroni et al., 2012)) by concatenating (*Conc*) the vectors of the input pair: $f(\vec{e}, \vec{c}) = Conc(\vec{e}, \vec{c}) = \langle e_1, \ldots, e_n, c_1, \ldots, c_n \rangle$. In Section 3.5.5, we will consider *difference* as the function f between $\vec{e}$ and $\vec{c}$ to generate the input representation. The motivation comes from the works of Mikolov et al. (2013c) and Roller et al. (2014) who report

its usefulness in representing analogy and taxonomy (through hypernymy) respectively, in distributional spaces.

3.5.1.1 Hyperparameters

We use mean cross-entropy as the loss function and *softmax* as the activation function for the output layer. All hidden layers use *tanh* as activation function. The number of units in each hidden layer of the NN models is optimized for each model separately. We consider the following values: 5, 10, 50–800 (step size 50). For NN-1HL, the optimal number of hidden units is 400. For NN-2HL, the number of optimal units in the first hidden layer is 250 and in the second hidden layer is 50.

All models are trained using Adadelta optimization (Zeiler, 2012) to a maximum of 2000 epochs with early stopping (we found that models typically converge at 50–100 epochs). To reduce overfitting, we introduce a dropout layer in front of each hidden layer with a standard dropout value of 0.5 (Baldi and Sandowski, 2013). We also experimented with an additional L_2 weight regularization, in the range $[10^{-2}, 10^{-6}]$, at the time of loss computation to further optimize over any parameters that might be outliers. However, given that dropout itself is known to induce regularization (Wager et al., 2013), an additional regularization step (with our range of values) did not lead to any change in results when compared to the models trained without L_2 regularization i.e., (dropout + L_2 regularization) = (dropout). We did not experiment with higher values of L_2 because that would lead to excessive smoothing of model parameters resulting in underfitting.

3.5.1.2 Baselines

We consider two baselines. A frequency baseline (BL_{freq}) assigns the positive class randomly with the true class probability: 50% for the balanced variants and 25% for the union variants (according to the dataset definition in Section 3.3.3). We do not consider a most frequent class baseline, because this baseline would not make any positive predictions on the

union variants, where the negative class is dominant. Instead, we consider a baseline that *always* assigns the positive class (BL_{pos}). This baseline should show strong results in our evaluation scheme (F_1-score on positive class; discussed next) since it always yields a perfect recall.

3.5.2 Evaluation

We use F_1-score on the positive data samples as our evaluation measure. While accuracy would be a simple alternative evaluation measure on the balanced variants of our dataset (described in Section 3.3.3): *Pos+NotInst-inClass, Pos+NotInst-global, Pos+Inst2Inst, Pos+Inverse*, F_1 generalizes well to minority-class setups, like our union variants: *Pos+Union-inClass, Pos+Union-global*.

Moreover, besides the aim of distributional modelling of instantiation, another aim is to publicly release a well defined Instantiation dataset. Due to this, we are not just interested in an empirical analysis of instantiation detection between entities and categories but also in the analysis of the adequacy of our negative examples; and, the F_1-score, by the way of its definition, captures both aspects.

We compute F_1-scores on individual folds and micro-average them. The difference between micro and macro average is negligible in our case, due to our equally sized test folds; see Section 3.3.4.

3.5.3 Main results

Table 3.4 reports the main results for the three models and the baselines. The top part of the table lists results for the balanced dataset variants, all of which have a positive class baseline of 0.67. The bottom part shows results for the *Union* variants, where the positive class is now a minority class, with a correspondingly lower positive class baseline of 0.40. The dataset variants are arranged in their expected level of increasing difficulty in classification from top to bottom. Indeed, we see a clear trend of decreasing F-Scores. Since the first two types of negative examples (*Inverse* and *Inst2Inst*) can be

TABLE 3.4: F_1 scores for instantiation detection, concatenated vectors.

Dataset	BL_{freq}	BL_{pos}	LR	NN-1HL	NN-2HL
Pos + *Inverse*	0.50	0.67	**0.96**	**0.96**	**0.96**
Pos + *Inst2Inst*	0.50	0.67	0.90	**0.91**	**0.91**
Pos + *NotInst-global*	0.50	0.67	0.55	**0.85**	0.82
Pos + *NotInst-inClass*	0.50	0.67	0.55	**0.70**	0.69
Pos + *Union-global*	0.25	0.40	0.55	0.75	**0.76**
Pos + *Union-inClass*	0.25	0.40	0.55	0.57	**0.63**

classified correctly solely by learning to distinguish categories and entities. Therefore, we concur that the high-level distinction between categories and entities can be made quite easily with standard vectors.

In contrast, *NotInst* (the setup where the confounders are also *entity – category* pairs, just not ones that exemplify the instantiation relation, like *Madame Curie – lawyer*) is a quite difficult setup for which the scores decrease markedly. Thus, it is the presence or absence of the specific instantiation relationship, over and above the general 'type signature', that is difficult to determine. In line with this interpretation, we see that for *NotInst* and *Union*, the *inClass* variants, where the confounders are more semantically similar to the correct answers, are much harder than the *global* variants, with models outperforming the baseline by at most 3 points (0.70 vs. 0.67).

LR beats the baseline for *Inverse* and *Inst2Inst* but not for *NotInst* or the *Union* variants. The failure of LR – a linear classifier – to properly learn instantiation is in line with the observations by Roller and Erk (2016) and Levy et al. (2015b), who found that linear classifiers are generally unable to learn semantic relations from vanilla vectors i.e., standard distributional representations generated from text.

In contrast, the NN models beat the baseline for all variants, even though they also see a decrease in performance for the harder variants. The benefit of the NN architecture compared to LR correlates strongly with the

difficulty of the task: For the easiest *Inverse* and *Inst2Inst* variants, LR and NN perform at par, while the hard *NotInst* cases see differences of more than 10 and 30 percent, respectively.

The two NN models perform similarly. On the easiest variants (*Inverse* and *Inst2Inst*), both models do equally well. The model with one hidden layer performs better on the balanced *NotInst* variants but is beaten by the two-layer model on the *Union* variants. This indicates that the combination of different confounders calls for a model that can perform substantial transformations of the feature space. Note also that all models perform worse on the *Union* variants than the average of their performances on the individual variants, indicating that the different kinds of confounders call for different transformations from the input through the hidden layers. In the remainder of this section we focus on the neural network model with two hidden layers because it is the best on the hardest variants.

We compute for significance by using bootstrap re-sampling (Efron and Tibshirani, 1994). The samples are selected with replacement on the test set and, the number of samples is equal to the size of the test set[14]. For each of our six datasets, we carried out significance tests between the two NN models. At $p = 0.01$, all differences among models are not significant.

3.5.4 Error Analysis

An error analysis of the predictions made by our best model, NN-2HL, shows that most errors stem from *semantic relatedness*, which is a well known problem arising due to the methodology of meaning construction by distributional models (Radovanović et al., 2010; Baroni and Lenci, 2011). As explained in Section 2.1, the meaning of a word is constructed from its contexts within a predefined window and any semantic association (similarity, dissimilarity, part_of, etc.) that a word has with its context is captured by the meaning representation. As a result, the words that share contexts might appear close (or similar) to each other in a distributional

[14] We follow the same strategy throughout our work, wherever this technique is employed.

space, despite their actual meanings being unrelated, distantly related or even opposite to each other. Consequently, distributional models are known to suffer from the inability to distinguish between semantic similarity and relatedness (Section 1.1.1).

More concretely, we find two distinct linguistic phenomena leading to errors affecting categories and two phenomena affecting entities. The first problematic phenomenon for categories is conceptual similarity. For example, *Edna Ferber*, a *writer*, is also predicted to be a *composer*, probably due to the similarities between the two artistic occupations (compare the use of English *to compose* for the production of both text and of music). The second problematic phenomenon is association. For example, the *Cheshire Cat* from the book *Alice in Wonderland* is correctly recognized as a *fictional character*, but it is also wrongly predicted to be a *writer*, presumably because it is often discussed in the context of literature and literary theory. Another example is the incorrect classification of *Henri Rousseau*, a *painter*, as a *writer* due to his work being discussed in literary contexts.

As for entities, the first source of errors is referential ambiguity: while many names are unambiguous when given in their full form, texts often use abbreviated versions that can refer to multiple entities (remember our earlier *Washington* example: the person, the state, or the city). For some names, even the full form is ambiguous, like for *Albert Smith*, the name of various politicians, cricketers and footballers. And also for example, *James Bond* was predicted wrongly to be a *poet*, presumably because its representation construction was also influenced by the co-occurrences of the last name *Bond* that is shared by many famous poets: *Ruskin Bond*, *Bruce Bond* and *Edward Bond*. In the absence of large-scale reliable co-reference resolution and entity linking methods, researchers need to resort to heuristics to map corpus occurrences onto concrete entities, and wrong decisions result in biased representations. This type of error is analogous to the pervasive issue of ambiguity in lexical semantics (Cruse et al., 1986), though from a referential perspective.

Occasionally, this problem also spills over into the category domain: when frequent entity names include the category name (*gulf – Gulf of Patras*), the vector for the category term *gulf* may be biased by occurrences that are really parts of entity names. Recall from Section 3.3 that our vocabulary is lower-cased. For English, this problem could be alleviated by not lower-casing, which however introduces large amounts of wrong ambiguity. For languages which capitalize all nouns, such as German, or logo-grammatic writing systems such as Chinese, the problem persists even when lower-casing.

The second source of errors for entities is *changes in the world over time*. For example, *Stagira* was a city in ancient Greece, and is recorded as such in WordNet. It was however not predicted as such, presumably because the newswire texts underlying the representations only refer to it as a ruins or more generally as a historical site. Similarly, *Etruria* used to be an independent *country* in pre-Roman times, but is not predicted to be a country, probably due to the predominance of occurrences related to when it was a region of the Roman empire. Thus, for entities we find a strong effect of referential aspects of meaning (McNally and Boleda, 2017; Westera and Boleda, 2019).

Occasionally, we also encountered errors due to missing relational information in WordNet. For example, *Richard Brinsley Sheridan* was both a *playwright* and a British member of parliament, i.e., a *politician*. Only the former relation appears in WordNet, but our model predicts also the second. Similarly, *Yalta* is only listed as a resort *city* on the Black Sea in WordNet, but our classifier adds the information that it is a *port*. These observations are in line with the known incompleteness of knowledge bases (as pointed out in Section 2.4.1) and the usefulness of distributional methods to complete them (Min et al., 2013).

TABLE 3.5: Effects of input function (concatenation vs. difference) on F_1 score for best model (NN-2HL).

Dataset	Conc	Diff
Pos + *Inverse*	0.96	**0.97**
Pos + *Inst2Inst*	**0.91**	**0.91**
Pos + *NotInst-global*	**0.82**	**0.82**
Pos + *NotInst-inClass*	0.69	**0.72**
Pos + *Union-global*	**0.76**	0.75
Pos + *Union-inClass*	0.63	**0.67**

3.5.5 Auxiliary Experiment 1: Effect of input representations

We next test a different function to combine the representations of the two input elements. Above, we only considered concatenation, which enables the model to freely combine the information of the two vectors, but does not make the dimension-wise correspondence between them explicit. We now use the difference function (*Diff*), defined as $f(\vec{e},\vec{c}) = \langle e_1 - c_1, \ldots, e_n - c_n \rangle$. This representation, inspired by Mikolov et al. (2013b) and Roller et al. (2014), explicitly links the information in the input pair by dimension. Thus, the difference input provides the model with a clearer notion of how the category and entity vectors are located *relative* to one another, at the loss of their *absolute* positions in the vector space. Note that: 1) *Conc* produces 2,000-dimensional while *Diff* produces only 1,000-dimensional input vectors; and, 2) we do not change the model hyperparamters (as described in Section 3.5.1.1) across different input representations because our aim here is not to optimize the classification objective but to compare the effectiveness of input representations with the rest of the environment being static.

As Table 3.5 shows, the input representation does not play a major role. Results across variants are generally very close, with one exception: *Diff* yields better results for the *inClass* variants, which are the most difficult.

This could be due to an advantage of the *Diff* vectors for the case of highly similar confounders: while *Conc* also contains the necessary information to make the decision, this information is distributed over components in the vectors whose correspondences by dimension the model must recover. Moreover, as Weeds et al. (2014) point out, the *Diff* representation removes the features common to both input vectors, which in our case are the generic features shared between the entity and the category. Thus, the resultant representation primarily contains entity specific (idiosyncratic) features that would enable the classifier to learn instantiation more effectively.

Once again, to test for significance of results between the input representations on the NN2HL, we use bootstrap re-sampling on the 6 datasets. The differences are not significant at $p = 0.01$

3.5.6 Auxiliary Experiment 2: Impact of Memorization

Section 3.3.4 discussed our strategy to counteract possible memorization effects. This section demonstrates that memorization issues, identified in previous literature for hypernymy detection (Roller and Erk, 2016), affect instantiation as well.

Table 3.6 shows the results of Experiment 1 in two different setups. The first setup (i.e., Filtering; three columns to the left) gives precision, recall and F_1-scores on the dataset built to counter memorization during training. Recall, from Section 3.3.4, that the counter-memorization filtering removes all lexical overlap between the training, validation and test sets. Therefore, the models are not given the chance to memorize the prototypical features of a specific entity or category that entail instantiation. For example, from the instantiation relation learnt between (*Einstein – scientist*) the model is not allowed to make predictions on items like, (*Mendel – scientist*) or (*Einstein – physicist*), where it might have memorized the features (even a partial subset) 'by heart'. Instead, we ask the models to learn instantiation from certain entity – category pairs and then generalize over completely unseen entities and categories. The second setup (i.e., No

Filtering; three columns the middle) gives results on the dataset created by random partitioning with potentially overlapping lexical elements. In this setup, an entity or a category in test data might occur in the training data as well, thus, increasing the chances of the model having memorized the features of entities and categories that contribute to instantiation. In the three right-most columns, we also show the difference in results between the two set-ups i.e., Δ = (No Filtering - Filtering). Note that, due to its lack of lexical overlap, we consider the Filtering setup to be a more realistic and accurate representation for instantiation modelling.

We first analyse by columns (setups) where we see a gradual decline in scores of both Filtering and No Filtering setups with the increase in the dataset difficulty (see Section 3.3.3). However, given the lexical overlap in the No Filtering setup, it is not surprising the models in this setup achieve substantially higher scores; resulting in an increase between 2 and 18% over the Filtering setup (see ΔF_1). This is arguably the result of the models successfully memorizing the characteristics of instantiation on already seen entities and categories at training. Next, we compare the ΔRec and ΔPrec columns. Since recall is the inverse measure of False Negatives and precision is an inverse measure of False Positives, ΔRec and ΔPrec allow us to compare the rate of change of the mis-classified positive examples vs. mis-classified negative examples between the two setups. Interestingly, except for *NotInst* datasets[15], the ΔRec values are greater than ΔPrec values. In fact, averaging ΔRec ($= 0.13$) and ΔPrec ($= 0.07$) tells us that the overall difference in recall is almost twice as compared to precision of the two setups. In other words, when model memorization is removed through lexical overlap in the Filtering setup, the models have a significant rise in mis-classified positive examples because they fail to sufficiently generalize over unseen data. This outcome fits

[15]In the negative examples of *NotInst* datasets, the antecedent entity is the same as that of the corresponding positive examples. Additionally, we ensure that the consequent of the negative example is a category which does not instantiate the entity, however, there is no straightforward method to ensure that entity and category are not semantically related. As pointed out in Section 3.5.4, this adversely affects the correct classifications of negative examples.

TABLE 3.6: Effects of Memorization on Precision, Recall and F_1-score for best model (NN-2HL)

Dataset	Filtering			No Filtering			Δ		
	Prec	Rec	F_1	Prec	Rec	F_1	ΔPrec	ΔRec	ΔF_1
Pos+*Inverse*	0.96	0.96	0.96	0.98	0.99	0.98	0.02	0.03	0.02
Pos+*Inst2Inst*	0.97	0.85	0.91	0.99	0.99	0.99	0.02	0.14	0.08
Pos+*NotInst-global*	0.80	0.85	0.82	0.88	0.92	0.90	0.08	0.07	0.08
Pos+*NotInst-inClass*	0.64	0.74	0.69	0.79	0.87	0.83	0.15	0.13	0.14
Pos+*Union-global*	0.80	0.73	0.76	0.89	0.89	0.89	0.09	0.16	0.13
Pos+*Union-inClass*	0.68	0.58	0.63	0.79	0.83	0.81	0.11	0.25	0.18

well into the initial observations we made in Section 3.4.1 that entities are widely dispersed within the semantic space as well as that there are regions occupied only by entities and not (their instantiating) categories. Thus, making instantiation detection a difficult task. We also compare these results with those from our hypernymy detection task to conclusively prove that instantiation and Hypernymy are two functionally distinct relations.

We now analyse by rows (datasets). In both setups, the *Inverse* dataset results in good performance which shows that instantiation is easy to detect when contrasted against asymmetry. Surprisingly, while the *Inst2Inst* dataset gives the best results in the No Filtering setup, we see a sharp drop in recall ($\Delta\text{Prec} = 14\%$) in the corresponding Filtering setup; although, the precision drops only slightly ($\Delta\text{Rec} = 2\%$). Basically, the model specifically mis-classifies positive examples. A plausible reason is that this dataset is designed to model instantiation vs. semantic similarity, and without memorization, the model is biased towards identifying semantic similarity when evaluated on previously unseen inputs. The variance in *Inst2Inst* is also mirrored in the *Union* datasets as these subsume the *Inst2Inst* dataset in addition to being generally tougher due to a higher number of negative examples per positive example.

To sum up, the differences that we see between the Filtering and No Filtering setups in Tables 3.6 reflect that memorization exists and that it can be countered with procedures that we follow. The differences between

the two setups are not significant at $p = 0.01$. We show that instantiation detection is a difficult task to model because of the nature of distribution of entities in the semantic space as well as the ability (or inability) of the model to learn effectively from confounders. Overall, our results without the lexical overlap are more meaningful with respect to modelling instantiation detection.

3.5.7 Conclusion

This work builds on a large body of previous work that has shown that distributional semantic representations are a reasonable proxy for conceptual aspects of meaning (Landauer and Dumais, 1997; Baroni and Lenci, 2010; Mikolov et al., 2013b, among many others), and on much less work on distributed representations of entities that has also given encouraging results (Mikolov et al., 2013c; Herbelot, 2015).

The main contributions of this work are: 1) we show the existence of general distinctions between entities and categories distributionally, which supports the postulation of this distinction in formal theories; 2) creation of a comprehensive dataset for instantiation, which enables future work in this area; and, 3) distributional modelling of the instantiation relation between entities and categories.

While the distributional distinctions between entities and categories are easy to recover, in contrast, the instantiation relationship proper is much more difficult, particularly when the confounders come from a similar domain. Even with a supervised approach using non-linear combinations of features (a two-layer neural network classifier) with strict controls for memorization, it is difficult to distinguish instantiation from mere semantic relatedness. The interference of semantic relatedness and similarity is common to lexical semantic phenomena (such as synonymy, hypernymy, or meronymy), as identified in previous work (Baroni and Lenci, 2011; Santus et al., 2014; Levy et al., 2015b). An additional difficulty in our case

is posed by referential aspects of meaning, such as the referential ambiguity of names and changes in the world over time.

Our setup is somewhat similar to fine-grained entity classification in NERC as well as NET in that we also experiment with Named Entities within the bounds of a classification setup. Ling and Weld (2012b) were among the first studies in this direction, performing NERC on a set of 112 classes. Abhishek and Awekar (2017) and Shimaoka et al. (2017) present neural architectures for fine-grained NEC applied to between 47 and 128 classes. However, our set of 577 classes is considerably larger than those normally considered in fine-grained entity classification. While our set of classes is considerably lesser than the 2,500 unique classes of Choi et al. (2018), their results are considerably lower as well and such works are found to suffer from a generally low recall i.e., high number of mis-classified positive examples.

Our systematic investigation, through analyses and experiments, shows valuable insights on the potential of current distributed representations to capture instantiation. Our insights also bring forward questions: *Can category representations constructed from entities also serve as a (better) proxy for representations constructed from text, with respect to relation modelling?* and *Is there a distributional distinction between instantiation and hypernymy – both of which differ in terms of the type of their word-pair antecedent?* We address these in the next chapter.

We leave two aspects to future work: 1) our experiments and analyses are based on one distributional space. While the space has a large coverage in terms of its elements and corpora, a comparison of instantiation detection across different semantic spaces will bring out more clarity on the distributional behaviour of instantiation; and, 2) Our current set of classes is biased towards the *person* domain owing to a similar bias in WordNet and the creation of a more balanced dataset from other varied resources will also add towards improvement in understanding of modelling of entities through distributional methods.

Chapter 4

Instantiation - Part II

In Chapter 3, we investigated the modelling of the lexical relation of Instantiation between an entity and a category that it instantiates.

In this chapter, we extend two strands from the previous chapter: first, we explore the possibility of optimizing instantiation detection by using an alternate representation for categories, where each category is a centroid computed from its instantiated entities; second, we model the lexical relation of hypernymy, through a hypernymy dataset that is comparable to our Instantiation dataset. We contrast the distributional behaviour and geometry of the two relations to show that the two distinct relations, not just in theory but also computationally, and should be treated as such.

4.1 Experiment 2: Instantiation Detection from Entity-Based Categories

One of our findings from the previous chapter is that Instantiation proper is tough to recover, in the sense that, some of the models have a tough time beating the baselines. We have suspected one reason (amongst others) to be that entities are quite distant from their instantiating categories in the semantic space and that inter-category semantic similarities are higher as compared to entity-category similarities (see Section 3.4.1 and Table 4.1). This raises a doubt on the suitability of using category representations constructed from standard distributional models for modelling instantiation. Subsequently, this brings forward a question: *Is there a better representation for categories which can supplement Instantiation detection?*

While investigating the distributional distinctions between entities and categories through a clustering analysis, we have already noticed some evidence for an alternate representation of categories. Our qualitative cluster analysis (in Table 3.3, Section 3.4.2.2) reveals entity organization through topicality (or, categories). We consider this as evidence towards the existence of prototypical information of categories encoded within the entity representations.

Thus, it is entirely plausible that an abstraction for a category can be derived from the entities it instantiates. More concretely, we propose that a concept-based[1] category representation (as defined in Section 3.3.1) can be substituted by a centroid-based category representation computed from its instantiated entities. We believe that this centroid-based representation will fare better in modelling instantiation.

Plan for Experiment 2 : In the following sections, we first discuss how our proposal correlates with other theories with respect to their interpretation of categories and subsequent categorization of entities. We then describe the datasets for the experiment and conduct a preliminary data analysis, similar to the one in Chapter 3. The analysis helps in grounding our proposal more firmly by observing the distributional geometry of centroid-based category representations. We then compare Instantiation detection through both centroid-based and concept-based category representations and discuss our empirical findings which will conclusively show the merit of our proposed approach.

4.1.1 Validation from Semantic Literature

Our proposal has a sound basis in cognitive semantics, resonating well with two theories in particular that assume categories to be inherently gradable:

[1]Concept-based representations are nothing but 'standard distributional word representations' constructed by a distributional semantic model over a corpus. Each word denotes a 'concept' and its meaning is constructed by the model through estimating distributions of words denoting other concepts.

1. **The exemplar theory** assumes that a category is represented by instances (or, instantiated entities) which have previously been encountered and stored in memory as instances of that category. A new item is categorized as an instance of a category to which it is sufficiently similar to or to its instances that exist in the memory (Medin and Schaffer, 1978). These instances, which are helpful in cognitively modelling categories, are also called *exemplars*. For example, a person might learn the category *scientist* by maintaining in their memory a collection of all the scientists they have encountered: *Albert Einstein, Issac Newton, Marie Curie, Louis Pasteur, Rosalind Franklin, Alfred Nobel, etc*. They might further learn to sub-categorize a scientist as a *physicist*, *chemist* or both based on the differences they observe between *Albert Einstein* and *Marie Curie*, the former a prominent physicist and the latter a physicist as well as a chemist.

2. **The prototype theory** assumes that the representation of a category is based on its *prototype or summary representation*. This prototype is represented by a set of features[2] where each feature is associated with a *degree* and *weight*. The degree is the ideal value of a feature with respect to the category and the weight is the relative importance of that feature as a constituent of the category. For example, the category *bird* has prototypical features like *size, wings, beak, claws, ability-to-fly and chirping* where the importance (weight) given to the features is unequal; *wings* and *ability-to-fly* are more important because they are unique amongst birds, whereas, *claws* might be common with other animals like, *crabs* – which makes the feature less important in categorizing a bird. An entity instantiates a category on the basis of similarity between the entity and the prototype of the category where similarity is computed as a function of the weights and values of the prototype features. Inversely, we can also say

[2]Note that while we use *features* to denote a basic set of attributes that define a category, in cognitive literature the term *dimensions* is more commonly used.

that an entity instantiates a category based on its distance with the prototype (Rosch, 1975).

Rosch and Mervis (1975) further state that the prototype of a category can be abstracted from the exemplars of that category that a person has previously experienced. According to them, some exemplars may be more representative or more typical of a category than others, for example, a *bird* might be better represented by a *crow* rather than a *penguin* due to some of its characteristics (like, *wings and ability-to-fly*) being more prominent than the other prototypical features it might share with the penguin. Additionally, exemplars are not associated to a category strictly through its defining features but instead they reflect more nearly a "family resemblance" structure or a prototype. While Reed (1972) assumes that every exemplar contributes to the prototype of a category, Rosch (1975) on the other hand assumes that only the most typical exemplars contribute to that prototype.

The main idea behind the cognitive semantic models is that of the *The Weighted Mean Hypothesis* (Wittgenstein, 2009), which states that the degree of typicality in a category can be given by computing a weighted mean of the features of that category. The features can be represented as lists (Hampton, 1979) or as vectors in conceptual spaces (Gärdenfors, 2004). The exemplar and prototype theories differ in their approach towards categorization, in the sense that, the former associates each category with many sets of features (observed from varied in- memory entities), whereas, the latter associates each category with one set of prototypical features. However, the salient point common to both, that we focus on, is that previously observed entities can be sufficient to represent the conceptual notion of the categories that they instantiate.

Our proposal is also partly rooted in the Distributional Inclusion Hypothesis (as defined in a footnote on Page 60): given that the contexts of a subordinate occur in the subset of the contexts of the superordinate, it

stands to reason that entities too would encode prototypical features of their instantiating category through shared contexts. And, while the distributional representation of one entity alone might not capture all the features, we assume that the aggregation of a set of entities might very well stand in lieu of the concept-based representation of categories.

An alternate approach to our proposal can be to model instantiation through exemplars of a category i.e., instead of using a category representation that is a centroid of entities, we can detect instantiation based on: 1) the averaged similarity that an entity exhibits with all the other entities that belong to a category; or, 2) the closest observed entity in the semantic space. In other words, instead of modelling instantiation as a classification task based on learning the interactions between an *entity – category* word-pair, we can model instantiation as a similarity-based task. However, we forgo the idea of similarity-based instantiation because exemplars models have been found to perform worse than the prototype models in lexical semantics (Sikos and Padó, 2019).

4.1.2 Datasets

We create a *new* Instantiation dataset that allows us to compare centroid vs. concept based representations in instantiation detection. We cannot use the Instantiation dataset that we previously used in Chapter 3 (described in Section 3.3) as-it-is because some categories in that dataset have a very small number of entities. Therefore, we reduce the positive datapoints in the original Instantiation dataset by keeping a threshold of a minimum of 5 word-pairs per category (or, 5 unique entities per category). This results in a decrease in the number of positive datapoints from 5,469 to 4,790 – a reduction of 12.41%; and, from (4750 unique entities, 577 categories) to (4180 unique entities, 159 categories)[3]. The dataset retains all the 7 ontological classes that were originally derived from WordNet.

[3]This reduced version of the Instantiation dataset is also available at `http://www.ims.uni-stuttgart.de/data/Instantiation.html`.

We use this reduced set of positive datapoints to generate the Instantiation datasets for the current experiment. First, with the same methodology that we have previously used in Section 3.3.3, we create four sets of confounders: *Inverse* (inducing asymmetry), *Inst2Inst* (inducing similarity), *NotInst-global* and *NotInst-inClass* (inducing incorrect categorization). The *Union* variants: *Union-global* and *Union-inClass* are the union of the negative confounders. We then generate our Instantiation datasets by pairing the six confounder sets with positive datapoints, resulting in four balanced (Pos+*Inverse*, Pos+*Inst2Inst*, Pos+*NotInst-global* and Pos+*NotInst-inClass*) and two imbalanced (Pos+*Union-global* and Pos+*Union-inClass*; with a 1:3 positive-to-negative ratio) datasets. Finally, we partition the datasets into training, validation and test sets (80, 10, and 10% respectively) and remove the lexical overlap (as described in Section 3.3.4) resulting in 83 training and test folds per dataset with more than 90% coverage of the original training data.

In terms of populating the datasets with vectors, we create two variants: *centroid-based* and *concept-based* (as defined in the beginning of Section 4.1). The entity and concept-based category vectors belong to the Google News distributional space mapped to Freebase identifiers (see Section 3.3.1). However, in the *centroid-based* Instantiation datasets, for every category, we compute a centroid vector by averaging the dimension-wise addition of all entity vectors which belong to that category *in the training data*:

$$\vec{c} = \frac{1}{n} \sum_{i=3}^{n} \vec{e}_i \tag{4.1}$$

Where:

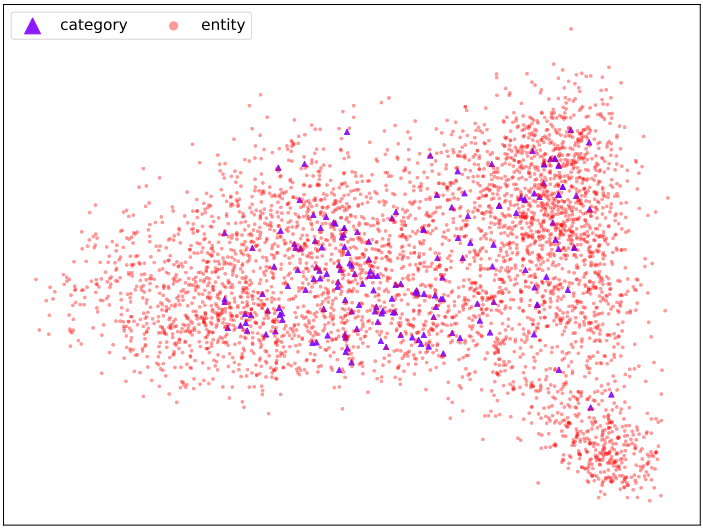
$\vec{c}$ = centroid vector of a category

$\vec{e}$ = concept-based entity vector

i = the set of entities that instantiate the category in the training data

Recall, from our discussion above, that we postulate that the resultant centroid vector is akin to the prototypical vector of the category. Our data

FIGURE 4.1: Entities and centroid-based categories in distributional space by first two PCA dimensions.



partitioning scheme along with the thresholding of 5 unique word-pairs per category ensures that every category has a minimum of 3 positive word-pairs in training data and 1 each in validation and test data. Since our training-validation-test sets are disjoint (i.e., without lexical overlap), for each entity in the test set, it is guaranteed that its vector was not used in the construction of any centroid vector.

4.1.3 Data Analysis

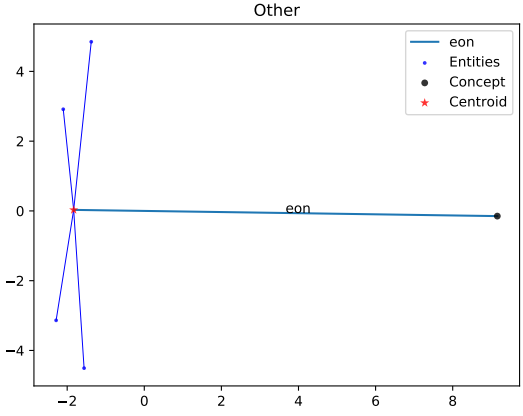
We carry out a preliminary data analysis on entities and their instantiating categories (both centroid-based and concept-based representations) picked from the positive datapoints of the Instantiation dataset.

We project the elements of the positive dataset to a 2D space by using PCA and map the entities and centroid-based categories on to a graph, as shown in Figure 4.1. We compare this figure with Figure 3.2, which represents concept-based categories in the distributional space. In both

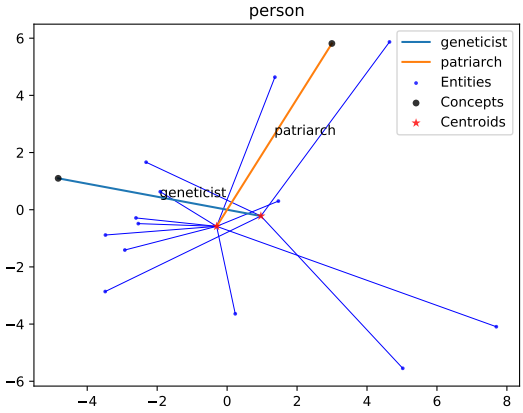
figures, entities show a similar distribution. However, in Figure 4.1, the categories are much widely dispersed and are no more congregated in a tightly-packed central radial structure. We believe this is because, within the centroid vectors, the idiosyncratic features are smoothed out and the typicality of category dimensions is retained to some extent. In fact, they are now located in the central regions of their corresponding domains (compare with Figure 3.3 for an overlay with the domain spreads). Although centroid-based categories still do not seem to be linearly separable (as seen in the graph) but, we take their wider dispersion as a signal of them being closer to their instantiated entities as compared to their concept-based counterparts.

To further validate the point we just made, we plot randomly selected categories (both the centroid and concept based representations) along with the the entities they instantiate. Figure 4.2 shows examples from two domains: *person* – the most populous and, *Other* – the most sparse, thereby, representing the two extremes of our Instantiation dataset. The conclusions that we make (in what follows) are generalizable because we observed identical behaviour displayed by categories and entities in the rest of the domains as well. In both Figure 4.2 (A) and (B), the entity vectors are shown as small dots, linked to their category centroid (shown as a red star). The concept-based category vectors are shown as a large dot. We see that the concept vectors (for the categories *eon, geneticist and patriarch*) are marginal, in terms of the overall data distribution, and that they live in a somewhat different part of the space; indicating that there is no simple relationship between entities and their instantiating categories. On the other hand (and as expected), the centroid-based representations of the categories are situated right in the middle of all the entities confirming that our alternate representation of a category is much better situated with respect to similarity with entities. Hence, its a better candidate for instantiation detection, as compared to its concept-based counterpart.

To generalize this conclusion we compute the average similarities between the entities, concept-based categories and centroid-based categories across the newly-designed reduced Instantiation dataset, as shown in Table



(A)



(B)

FIGURE 4.2: Entities, centroid vectors, and concept vectors for the most sparse (*Other*) and the most populous (*person*) domains of the Instantiation dataset

TABLE 4.1: Cosine similarities between entities, concepts and centroids (means and standard deviations).

	Across-categories	Within-categories
Entities	0.05 (0.07)	0.22 (0.11)
Concepts	0.06 (0.06)	–
Centroids	0.20 (0.12)	–
Entity-Concept	0.04 (0.05)	0.16 (0.09)
Entity-Centroid	0.10 (0.09)	0.55 (0.11)
Concept-Centroid	0.08 (0.07)	0.29 (0.14)

4.1. Here, *Across-categories* compares an entity/category to all other entities/categories, for example, *Madame Curie* vs. *George Washington*, *Albert Einstein*, *Mumbai*, *Nile*, etc., and *scientist* vs. *president of the United States*, *city*, *river*, etc. On the other hand, *Within-categories* restricts comparison to a single category: *Madame Curie* vs. *Albert Einstein*, *Mendel*, etc. We see that the centroid-based representation appears to be promisingly well-behaved. While the average similarity of each entity to its category is 0.16 (e.g., *Madame Curie – scientist*), the average inter-entity similarity within categories is 0.22 (*Madame Curie – Einstein*, *Mendel*, ...). The similarity of entities to the centroid is 0.55; a high similarity is to be expected here simply from the definition of centroid. The numbers show that distributional representations of entities belonging to a certain category are more similar to each other than to their category (0.22 vs. 0.16). Moreover, the fact that the average similarity between categories and centroids is higher than between entities and their categories (0.29 vs. 0.16) suggests that centroids make good category representations, indeed smoothing out idiosyncratic differences between the entities as expected.

4.1.4 Experimental Setup

The Instantiation datasets in this experiment differ with the datasets of the previous chapter only by the way of a lower number of unique categories

TABLE 4.2: F_1 scores for instantiation detection, concept-based vs. centroid-based category representation

		Concept-based		Centroid-based	
Dataset	BL_{Pos}	NN-1HL	NN-2HL	NN-1HL	NN-2HL
Pos + *Inverse*	0.67	0.98	0.97	**0.99**	0.93
Pos + *Inst2Inst*	0.67	0.90	0.91	**0.92**	0.86
Pos + *NotInst-global*	0.67	0.85	0.84	**0.90**	0.89
Pos + *NotInst-inClass*	0.67	0.71	0.67	**0.77**	0.74
Pos + *Union-global*	0.40	0.73	0.76	**0.84**	0.75
Pos + *Union-inClass*	0.40	0.51	0.65	**0.75**	0.68

as well as their different representations but, not in terms of their design.

Therefore, in this chapter as well, we model Instantiation detection as a supervised binary classification task. Our models to train and evaluate the datasets are the same as in the previous experiment: two feed-forward Neural Networks with 1 hidden layer (NN-1HL) and 2 hidden layers (NN-2HL); we focus on the neural models on account of their better performance as compared to the logistic regression classifier. We use the same hyper-parameters, baselines and evaluation metric as well (see Section 3.5.1 and 3.5.2 for details).

4.1.5 Results and Discussion

Table 4.2 shows the results. The positive class baseline performs as before, since the class distributions do not change. The results using a concept-based embedding to represent categories (middle columns) can be compared with the corresponding numbers in Table 3.4. The results are rather similar, around 1–2% higher than before. This indicates that the reduced dataset is comparable in difficulty to the original dataset from Experiment 1.

Comparing to the centroid-based approach (right-hand columns), we see that the centroid-based approach outperforms the concept-based approach on all variants. A notable difference from Instantiation detection results in

Chapter 3 is that this time the neural network with a single hidden layer performs best (see column NN-1HL), consistently across all variants. We take this as evidence that the centroid-based representation requires less transformation of the input, compared to the concept-based representation. This is consistent with the analysis in Section 4.1.3, which showed that entities are closer in space to their category centroid than to their category denoting a concept.

As for the different dataset variants, Table 4.2 shows that the centroid-based representation confers significant gains for the hardest variants i.e., the last four datasets in the table. On *Inverse* and *Inst2Inst*, we obtain only a marginal improvement on performance, which makes sense because these variants require the model to distinguish between entities and categories and deal with asymmetry and similarity as confounders. The centroid-based representation does not provide further help in this case, presumably for the same reason as above (in the concept-based representation, categories are more different than entities, and consequently, it is easier to distinguish them from entities). It is however noteworthy that despite this reasoning, the results of the centroid-based models do not suffer.

In contrast, representing categories as centroids of entities yields big gains for the *NotInst* and *Union* cases. *NotInst-global* and *Union-global* reach 0.90 and 0.84 performance, respectively, making them almost comparable to the easier settings (*Inverse* and *Inst2Inst* have 0.99 and 0.92 F_1-score). The hardest setups, *NotInst-inClass* and *Union-inClass*, improve by 6 and 24% to 0.77 and 0.75 F_1-score, respectively.

At $p = 0.01$, for the centroid-based setup the differences are significant with NN-1HL being the better model as compared to NN-2HL. The differences between the models of the concept-based setup are not significant.

Thus, we conclude, that in the centroid-setup recovering the Instantiation relation is much easier than from concept-based representations.

4.2 Experiment 3: Instantiation Vs Hypernymy Detection

We now turn our attention towards our second objective of comparing instantiation detection vs. hypernymy detection. Both relations are comparable on the grounds of being a subordinate-superordinate taxonomical lexical relation but different in their antecedents where Instantiation has an entity whereas hypernymy has a category; for example, (*Lassie – dog*) represents instantiation in our Instantiation dataset and, (*dog – animal*) represents hypernmymy in our Hypernymy dataset. Due to this, as proposed in Section 1.3, instantiation detection should be fundamentally different from the hypernymy detection as a task; which we aim to show through this experiment.

Plan for Experiment 3 : In the following sections, we first discuss in detail about the motivation and methodology followed in creating the Hypernymy datasets. We then briefly describe the experimental setup which, in essence, mirrors the instantiation detection setup described in Section 3.5.1. We then discuss the results of modelling hypernymy detection as a task and, follow it up with a comparison between the results of instantiation detection vs hypernymy detection.

4.2.1 Data: Instantiation and Hypernymy

Within the scope of this experiment, our objective is not to optimize relation classification but to compare the modelling performance of the two relations. Therefore, our top priority is to achieve semantic equivalence (atleast as much as possible) between the Instantiation and the Hypernymy datasets, and consequently avoid any lexical biases that may arise from the dataset design. Since we already have one of the datasets available at hand (Instantiation), we will derive the second dataset (hypernymy) through the first.

We use the Instantiation dataset previously used in Chapter 3 without lexical overlap i.e., the one with 5,469 positive datapoints containing 4,750 unique entities and 577 categories. We do not use the subset (reduced version) of the Instantiation dataset defined in the previous experiment (Section 4.1.2) due to a comparatively lower number of categories which will also adversely affect the size of the Hypernymy dataset that we create in the following section.

Due to the desired semantic equivalence, we do not use the standard benchmark Hypernymy datasets, like BLESS (Baroni and Lenci, 2011) and ENTAILMENT (Baroni et al., 2012) which incidentally also have a smaller number of categories (see Section 3.2.2.1 for details). We also do not consider creating a dataset simply by directly extracting hypernymy word-pairs from lexical resources and large scale taxonomies, as done by Shwartz et al. (2016). Instead, we choose build a comparable hypernymy dataset from scratch.

4.2.1.1 The Hypernymy Dataset

Positive Datapoints: We start by extracting the set of 577 categories from the positive examples of the instantiation dataset. For each category, we use the WordNet *hypernyms* function to extract all possible hypernyms of that category. Similar to our instantiation mapping procedure in Section 3.3.2, we map the hypernyms to Freebase identifiers. We keep the match with the longest token sequence and discard the rest. Pairing each category with its hypernym results in a set of 577 datapoints where there is a one-to-one correspondence with the categories in the Instantiation dataset. For example, for the category *physicist* in the instantiation datapoint (*Albert Einstein – physicist*), we create a positive datapoint (*physicist – scientist*) reflecting the hypernymy relation. Figure 4.3 shows a graphical description of the hypernymy word-pair extraction process.

While we have achieved balance in terms of semantic equivalence, this 577 set of hypernymy datapoints is a relatively small-sized dataset which

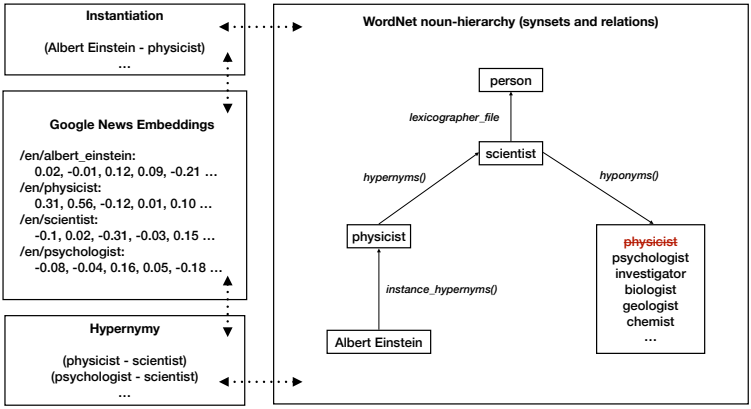


FIGURE 4.3: Hypernymy dataset construction from Instantiation dataset and WordNet noun-hierarchy with mapping to Google News targets.

poses a concern: the supervised model may not have sufficient examples for generalized learning. Therefore, for each datapoint in this set, we also extract the co-hyponyms of the hyponym using WordNet i.e., for (*physicist – scientist*) in the above example, we extract all the co-hyponyms of *physicist* which gives us: (*psychologist – scientist, investigator – scientist, biologist – scientist, geologist – scientist, . . .*). Note that we ensure that the hypernym (or, the superordinate) is the same for all co-hyponyms and is compulsorily their immediate taxonomical parent, like *scientist* in the above example. Lastly, we associate each datapoint with an ontological class extracted from WordNet LEXICOGRAPHER file labels. After combining these datapoints together into a set, we finally get a hypernymy dataset of positive examples totalling 2,281 unique datapoints (word-pairs), belonging to 7 ontological classes, with 2,134 unique hyponyms and 378 unique hypernyms.

A statistical summary of the positive datapoints can be seen in Table 4.3. The hypernymy dataset is divided into the same ontological classes as seen in the instantiation dataset (Table 3.1); this is expected because the two datasets are linked through instantiation categories (as described above)

TABLE 4.3: Positive datapoints for hypernymy: Statistics and examples by ontological class.

Ontological Class	#Datapoints	#Hyponyms	#Hypernyms	Example
person	1267	1166	138	chemist – scientist
location	181	171	28	city – administrative division
object	136	132	24	river – stream
artifact	121	119	42	fortress – defence
communication	98	95	23	gospel – religious text
act	69	68	18	siege – blockade
other	409	383	105	scandal – trouble
Total unique	2281	2134	378	—

which serve as hyponyms in the hypernymy dataset. We see that the hypernymy dataset is also biased towards the *person* domain. This is because categories in the *person* domain are linked to 'professions/services/jobs' and these would tend to have a high number of co-hyponyms. On the other hand, the *location* and *object* domains have substantially reduced as compared to Table 3.1 and the number of co-hyponyms found in these domains is also quite low.

Confounders: To create the confounders, we adopt the strategy described in Section 3.3.3. We first create 4 sets of confounders (Inverse, *Hypo2Hypo*, *NotHyper-global* and *NotHyper-inClass*). For each positive hypernymy relation datapoint made of a *hyponym (hypo) – hypernym (hyper)* word-pair, we create the following negative examples:

1. *Inverse*: Swap the positions of hyponym and hypernym, yielding $(hyper, hypo)$.

2. *Hypo2Hypo*: Replace the correct hypernym by a different random hyponym $hypo'$ of the same ontological class, yielding $(hypo, hypo')$.

3. *NotHyper-global*: Replace the correct hypernym $hyper$ by a random wrong hypernym $hyper''$, from the global distribution of hypernyms, yielding
$(hypo, hyper'')$.

4. *NotHyper-inClass*: Replace the correct hypernym $hyper$ by a wrong hypernym $hyper'$, this time sampling from the same ontological class, yielding $(hypo, hyper')$.

We then create two more confounder sets by combining the first three $(1+2+3)$ and the first two with the last $(1+2+4)$ confounder sets which we referred to in the previous chapter as the *Union* variants – *Union-global* and *Union-inClass*, respectively. Next, we generate the hypernymy datasets by combining each confounder set with positive datapoints, resulting in four balanced (Pos+Inverse, Pos+*Hypo2Hypo*, Pos+*NotHyper-global* and Pos+*NotHyper-inClass*) and two imbalanced datasets (Pos+*Union-global*, Pos+*Union-inClass*). We partition the datasets into training, validation and tests sets (80,10,10%) respectively. We also remove the lexical overlap by creating train and test fold in which the elements (hyponyms and hypernyms) occurring in a test fold are removed from the corresponding train fold. Since the overall number of datapoints is lower (as compared to the Instantiation datasets), we require only 40 train-test folds to create a coverage of around 90% of the original training data in the train folds. We populate the datasets with vectors from the semantic space described in Section 3.3.1 by matching the dataset elements with the vector labels.

As a result of our dataset definitions, the six hypernymy datasets are comparable to their corresponding counterparts in the Instantiation dataset.

4.2.2 Experimental Setup

Since we want to compare the distributional behaviour of *instantiation detection* vs. *hypernymy detection*, we train and evaluate the hypernymy datasets on binary classifiers that use the same model architectures: Logistic Regression (LR), feed- forward Neural Network with 1 hidden layer (NN-1HL) and 2 hidden layers (NN-2HL)[4]. Each input to the classifiers is

[4]We also experimented with NN models upto 4 hidden layers but, just was the case in instantiation detection, we observed a decline in the model performance beyond the 2 layered models.

a function/operation on the antecedent and consequent (hyponym and hypernym, respectively) of a datapoint. We experiment with two functions, *concatenation (Conc)* i.e., vector concatenation and *difference (Diff)* i.e., dimension-wise component subtraction (explained in Section 3.5.5). *Conc* and *Diff* both performed similarly on instantiation detection and we would like to compare their effect on the hypernymy detection models as well. Moreover, *difference* has been shown to work more effectively in modelling hypernymy as compared to any other input representation (Roller et al., 2014).

We tune these models on the same range of hyperparameters, as described in detail in Section 3.5.1.1 and, use the same baselines – BL_{freq} and BL_{pos} (Section 3.5.1.2) – as well as evaluation metric: F_1-score (Section 3.5.2).

4.2.3 Results

We divide this section into two sub-sections. First, we will discuss the results of hypernymy detection, as a task, based on the experimental setup described above. In the second sub-section we will compare the best models of instantiation and hypernymy detection tasks.

4.2.3.1 Hypernymy Detection

Table 4.4 shows the F_1-score results of modelling hypernymy detection across various setups on *Concat* and *Diff* input representations. The leftmost column shows the datasets which are listed in order of increasing difficulty (as explained in Section 3.3.3). For each dataset, the first two columns show the two baselines (BL_{freq} and BL_{pos} respectively) followed by the LR model (in the middle). The last four columns show the results on the neural networks with 1 hidden layer (NN-1HL) and 2 hidden layers (NN-2HL). NN-1HL has 400 hidden units and in the NN-2HL, the first and the second layers have 250 and 50 hidden units, respectively. For each dataset, the topmost score is highlighted.

TABLE 4.4: F_1 scores for Hypernymy detection on *Concat* and *Diff* input representations (with memorization filtering).

Dataset	BL_{freq}	BL_{pos}	LR		NN-1HL		NN-2HL	
			Concat	Diff	Concat	Diff	Concat	Diff
Pos+*Inverse*	0.50	0.67	0.69	0.69	0.67	0.67	**0.70**	**0.70**
Pos+*Hypo2Hypo*	0.50	**0.67**	0.48	0.47	0.38	0.54	0.48	0.56
Pos+*NotHyper-global*	0.50	0.67	0.58	0.54	0.67	**0.71**	0.64	0.70
Pos+*NotHyper-inClass*	0.50	0.67	0.46	0.50	0.59	0.65	0.51	**0.68**
Pos+*Union-global*	0.25	0.40	0.28	0.27	0.08	0.27	0.21	**0.43**
Pos+*Union-inClass*	0.25	**0.40**	0.21	0.22	0.10	0.23	0.13	0.38

We see that there is no model that consistently outperforms the baselines, specially BL_{pos}, which indeed is a tough baseline to beat. In fact, BL_{pos} reports the highest score in two out of six datasets. This shows that modelling hypernymy detection, as a task, is non-trivial for all models across our datasets. We believe that there are two factors which together bring about this difficulty: 1) as shown in Section 3.4.1, categories appear to be quite close to each other in our semantic space and detecting hypernymy amongst highly similar elements is challenging; and, 2) recall that the hypernyms in the dataset are compulsorily the immediate parents of their hyponyms, and perhaps the distributional inclusion hypothesis (or generally, the specificity of hyponymys) does not hold perfectly and reflect effectively in the distributional features; we did not consider a greater hierarchical distance between the hyponyms and hyperyms because we want the hypernymy dataset to be analogous to the instantiation dataset.

The LR model beats BL_{pos} only in the *Inverse* dataset and in the rest, the scores are substantially lower; in the tougher datasets the LR scores are even lower than BL_{freq} in *Union-inClass*. This shows that distinguishing hypernymy is not a linear problem. Naturally, the non-linear models perform better, but only marginally. The NN-1HL model scores equal to BL_{pos} in *Inverse* and is 4% better in the *NotHyper-global* dataset. NN-2HL, which is our best model as well, outperforms BL_{pos} in all but *Hypo2Hypo* and *Union-inClass* datasets, however, the improvement ranges between 1 to

3% only. *Hypo2Hypo* scores seem to suffer across all models. A classifier trained on this dataset should be able to distinguish hypernymy vs. high semantic similarity which, we believe, the models fail to learn due to the potentially high similarity in the positive datapoint elements as well. The overall low and near-baseline scores indicate that the models are unable to generalize over the distinctions between hyponyms and hypernyms as well as hypernymy and non-hypernymy when presented with data that counters model memorization. We tested for significance at $p = 0.01$ between the three models (pairwise) through bootstrap re-sampling, and we do not find the differences to be significant.

Our hypernymy detection results are in-line with those reported in previous work which uses datasets analogous to our *NotHyper-global* i.e., where the word-pairs in the negative examples instantiate a non-hypernymy relation and, the datasets are designed to counter memorization effects through a technique similar to ours. For instance, Shwartz et al. (2016) report an F_1-score of 0.70 on a dataset of about 28K hypernymy word-pairs. We too report an F_1-score of 0.70 but, on a comparatively smaller dataset. However, while similar model performances on similarly designed datasets do indicate towards a general difficulty of the task, the results are not directly comparable in terms of the dataset and semantic space; we leave it to future work to examine the influence of these factors.

4.2.3.2 Instantiation vs. Hypernymy Detection

Table 4.5 compares the F_1-scores of the best models of instantiation detection (on the left) vs. hypernymy detection (on the right); the best model being NN-2HL in both cases. We report scores on all the six datasets over *Concat* as well as *Diff* input representations. Recall from above, that by design, the instantiation and hypernymy datasets mentioned in the same row are semantically comparable. Through these datasets, the classifiers learn to distinguish instantiation of a semantic relation vs asymmetry (*Inverse*),

TABLE 4.5: F_1 scores for Instantiation vs. Hypernymy detection on the best model (NN-2HL) with *Concat* and *Diff* representations.

Instantiation			Hypernymy		
Dataset	NN-2HL		Dataset	NN-2HL	
	Concat	Diff		Concat	Diff
Pos+*Inverse*	0.96	**0.97**	Pos+*Inverse*	**0.70**	**0.70**
Pos+*Inst2Inst*	**0.91**	**0.91**	Pos+*Hypo2Hypo*	0.48	**0.56**
Pos+*NotInst-global*	**0.82**	**0.82**	Pos+*NotHyper-global*	0.64	**0.70**
Pos+*NotInst-inClass*	0.69	**0.72**	Pos+*NotHyper-inClass*	0.51	**0.68**
Pos+*Union-global*	**0.76**	0.75	Pos+*Union-global*	0.21	**0.43**
Pos+*Union-inClass*	0.63	**0.67**	Pos+*Union-inClass*	0.13	**0.38**

vs. high semantic similarity (*Inst2Inst* and *Hypo2Hypo*) and vs. any other type of relatedness (or generally, a non-hypernymy relation).

The table shows that instantiation detections is comparatively an easier task than hypernymy detection; the instantiation scores range between 0.63 to 0.97 whereas hypernymy scores range between 0.14 to 0.70. In other words, it is easier to learn the distinctions between entities and categories as compared to hyponyms and hypernyms. As explained in Section 3.5.5, the effect of input representations *Concat* and *Diff* is fairly similar in instantiation with a variance of 1–4% between the two, however in hypernymy, the modelling performance improves by 6–25% when *Diff* is used for input representations. The higher performance of *Diff* is because the dimension-wise difference captures the degree of distributional inclusion on that dimension[5]. This fact brings out the most critical distinction between the two relations, which incidentally also shows our hypothesis, that the two relations are semantically different, to be correct. More concretely, while both relations hold a subordinate-superordinate relationship between

[5]Interestingly, Roller et al. (2014) state that difference-based inputs work better on linear classifiers as compared to non-linear ones, which is not true in our case (see Table 4.4). The input representation they use is $(Diff)^2$ – squared difference vector, which we also experimented with as an alternate input representation, however, we found *Diff* to perform better than $(Diff)^2$ on all our models (including LR).

the antecedent and the consequent, distributional inclusion is more true for hypernymy as compared to instantiation – as can be seen with the success of the *Diff* representation. In Chapter 3, we have already reported no significant differences on varied input representation for Instantiation. However, in the case of hypernymy, we found the differences between the input representations to be significant for the last three datasets of Table 4.5; where *Diff* is better than *Concat*.

When we compare the two relations by datasets, we see that the instantiation scores reflect the difficulty of the datasets i.e., there is a gradual decline in the scores (we observe this consistently in the previous two experiments as well). However, the scores of the hypernymy models do not reflect this graded difficulty. The scores of *Inverse* fall between the *NotHyper* variants and *Hypo2Hypo* is significantly lesser than both. The two datasets are termed as 'easy' because they are designed to leverage the semantic similarity (or, dissimilarity) between the subordinate and superordinate terms to detect the instantiation of a semantic relation, as seen in the instantiation results on the left. We attribute the low scores, specifically on these two datasets, in hypernymy to the fact that the hypernyms in our dataset are the immediate parents of the hyponyms. Consequently, instead of becoming a leveraging factor, the high semantic similarity between the two terms in the positive as well as negative examples adversely affects the ability of the models to detect hypernymy. The *Union* variants continue to be the worst performers in both datasets owing to a larger level of difficulty due the three different types of confounders being used in those datasets; note that this is considerably more complicated than having multiple negative examples of the same type.

To sum up, we compare the results of modelling instantiation vs hypernymy and our findings show that without memorization, hypernymy is tougher to model as compared to instantiation. And, by observing the distributional behaviour of instantiation and hypernymy, we conclusively show that the two relations are semantically quite distinct.

4.3 Conclusion

In the previous chapter, we concluded our instantiation modelling experiment by observing that instantiation proper difficult to recover. In this chapter, we start by exploring the possibility of optimizing instantiation by using centroid-based category representations. We show that the centroid-based category representations are not only semantically closer to their instantiating entities but that they also retain sufficient prototypicality of the categories they represent; ultimately leading to an improved modelling of instantiation through a comparatively simpler model architecture.

It would be interesting to see if a similar approach can be employed to model other lexical relations or even linguistic phenomena that involve categories, for example, hypernymy or entailment. The motivation comes from Rosch et al. (1976), who proposed the *basic-level* hypothesis which states that in a taxonomy there is a basic level of abstraction at which categories carry the most information. For example, in the bottom-up hierarchy of (*kitchen-chair – chair – furniture*), *chair* is the most informative level of categorizing an object as a chair because its subordinate and superordinate share features with other categories and hence, are less informative. However, it would be extremely challenging to identify this basic level for different categories and we leave the centroid-based hypernymy detection to future work.

The second part of this chapter has its motivation in resolving a claim that we make in the beginning of the thesis that instantiation, by its definition, is semantically different than hyponymy. This difference should reflect when the two relations are modelled computationally as well. Through our experiments we conclusively show our supposition to be true. The distributional behaviour of instantiation is different from that of hypernymy. In fact, while we stated that instantiation is difficult to recover, we found that hypernymy is even tougher to model, given that the two datasets are semantically comparable and restrict model memorization. We attribute this to the high semantic similarity between the hyponyms and hypernyms

in our dataset. Our input representation (*Diff*), indeed improves the model performance but, does not alleviate the problem completely. This too makes a case for finding an alternate representation for hypernyms (as suggested above).

Our choice of similar setups to model instantiation and hypernymy is deliberate and through this experiment we bring to light the fact that the conclusions drawn for categories (concepts) do not necessarily hold true for entities (instances). Overall, a deeper understanding of entities is warranted (in terms of the information they encode) so that they can be modelled more effectively in linguistic phenomena and applications that require them.

Chapter 5

Fine-grained Attribute Prediction - Experiment I

In Chapter 3 and 4, we empirically established and discussed the existence of distributional distinctions between entity and category representations. We explored their distributional geometry in the semantic space inhabited by both as well as by analysing the distributional behaviour of two similar lexical relations: *Instantiation* and *Hypernymy*, that differ in terms of their usage of entities and categories in the sub-ordinate position.

In this chapter, we focus on the specific information that we believe is implicitly included in entity vectors, which makes them distinct from categories. In Chapter 1 (Section 1.2.1), we postulated one main reason for this distinction to be the encoding of fine-grained real-world information into the entity representations by the distributional semantic models. Therefore, in what follows, we address the second research question raised in Section 1.3: *Is fine-grained information encoded within distributional representations?* We back our hypothesis with empirical evidence from our experiments. Furthermore, through detailed analyses, we also highlight what type of fine-grained information is easy (or, difficult) to extract and what are the factors that potentially affect fine-grained information prediction.

5.1 Background

Distributional Semantic Models (DSMs) excel at capturing fuzzy, graded aspects of meaning by observing (or, estimating) the distribution of a word with other words in context. Approximating semantic similarity by graded geometric distance in a vector space is an effective strategy to address the many linguistic phenomena that are better characterized in gradient rather than discrete terms, such as selectional preferences, semantic priming and, classifying relations between word-pairs (Padó and Lapata, 2007; Baroni and Lenci, 2010; Erk et al., 2010). Consequently, DSMs can be used to predict, amongst many other things, that *dog* is more similar to a *cat* than a *stone*. They also predict generic properties of concepts, such as mammals being typically furry (Baroni et al., 2010).

This notion of *graded similarity*, that works well for categories, also percolates to entities. For instance, and in line with the example above, we can say that *Germany* is more similar to *Netherlands* than to *Italy*. This is because, as highlighted in Section 1.1.2, the 'distributional meaning' of any word is merely an approximation of the actual meaning constructed from an aggregation of its contexts and, different word types (for example, *categories* denoting concepts and *entities* denoting instances of concepts) are treated *at par*.

However, not all aspects of human semantic knowledge are satisfactorily captured in terms of fuzzy relations and graded similarity. In particular, our knowledge of the meaning of words denoting specific entities involves a number of *hard facts* about the referents they denote. For example, entities (in contrast with categories) are also a rich source of world knowledge which is typically of a precise and fine-grained nature (Limaye et al., 2010). Since this information semantically grounds an entity, we also call it *referential information*. For example, the entity *Germany* has a *GDP of 3.6 trillion euros* as well as *GDP per capita of 44.5 thousand euros* and is located in the *continent of Europe*.

Fine-grained / referential information is found to occur around entity

mentions in text and, due to its high relevance in NLP tasks centered around reasoning and inferencing, it is most prominently seen in structured knowledge bases such as Freebase or Wikidata (Bollacker et al., 2008; Vrandečić and Krötzsch, 2014) (see Section 2.4 for a detailed overview). Knowledge bases store fine-grained information in a more structured and machine-readable format as attribute-value pairs of entities, for instance, the facts mentioned above about *Germany* can be stored as: *(GDP - 3.6 trillion Euros), (GDP per capita - 44.5 thousand Euros)* and *(Continent - Europe)*. Despite many attempts (partly covered in Section 5.2), there is no way to automatically construct large knowledge bases from unannotated data. Therefore traditionally, knowledge bases were created and curated manually or through pattern based extraction techniques (see Section 2.4). However, as mentioned in Section 2.4.1, knowledge bases suffer from *incompleteness* due to the ever growing nature of world-knowledge. The task of curating perennially evolving knowledge bases manually (or, even semi-automatically) is an undertaking that requires tremendous effort, and therefore, has become a research area in its own right and, it is called *Knowledge Base Completion (KBC)*. In recent times, exploiting corpus evidence through distributional semantics for KBC has gained popularity (Buitelaar and Cimiano, 2008). This is because DSMs can be induced in an unsupervised manner from large amounts of unannotated data and, allow massive harvesting of word meaning representations.

While distributional methods are being increasingly used for KBC, distributional semantics has skirted the issue of reference until now; with the notable exception of Herbelot and Vecchi (2015) (discussed in Section 5.2). One main reason is that fine-grained information, which denotes a referent and typically has a precise nature (as defined above), is considered to be difficult to express in terms of linear algebraic frameworks. Moreover, recall from above that the notion of meaning in distributional semantics is aggregative, therefore, it is assumed to lack the element of preciseness (Murphy, 2004; Baroni et al., 2014a). With this in consideration, for a while it was also assumed that distributional methods could not predict

fine-grained information. But, their successful application in KBC shows that this assumption is not categorically true.

The fact that fine-grained information can also be extracted from fuzzy conceptual spaces, is also corroborated by empirical studies on spatial biases in cognition. In cognitive literature, a prominent line of thought claims that concepts arise out of languages and that conceptual structures and semantic structures are closely coupled (Dennett, 1993). It has also been observed that human spatial biases related to geographical locations are influenced through a combination of beliefs (both accurate and inaccurate) arising from cultural and socio-economic factors perceived about those locations or their superordinate regions. For example, Friedman et al. (2002) report that Texans locate Canadian cities closer to the US border with greater accuracy as compared to Mexican cities (despite the proximity of Texans to the latter) presumably on grounds of greater cultural similarities with the former. They also place Southern US cities further south than they really are, possibly based on a cognitive weather-to-latitude heuristics – the warmer the place, the close to equator it must be.

These beliefs (or, broadly-conceptual factors) are exactly the sort of information we would expect DSMs to capture. Indeed, Louwerse and Zwaan (2009) showed that coordinates of geographical locations represented through a Latent Semantic Analysis space (constructed from newspaper text) have a strong correlation with the actual latitude and longitude of those locations. Their results show that: 1) similar locations share similar contexts by the way of these locations typically being discussed in text in conjunction; and, 2) that language encodes geographical information (although not necessarily in terms of spatial descriptions) some of which can be computationally estimated, like location and population, to near-human accuracies.

They also arrive on similar conclusions (as previous works) with regard to biases that: geographical belief biases in participants has an affect on their geographical judgements. This brings to light another consideration

that biases do not just facilitate (like above) but also hinder human judgements. If the postulation made by Landauer and Dumais (1997) is true, that distributional models mirror cognitive models of human learning, then DSMs will both excel as well as fail at estimating information due to the conceptual biases that arise from contexts in a corpus. At the time of this work, there were no prior studies that extensively explored in this direction.

Thus, from a distributional point-of-view there are questions, ultimately subsumed by our primary research question, that remain empirically unanswered or under-analyzed. For instance, given the distributional meaning representation of an entity: 1) *Can we make reasonable predictions for the entire range of attributes of an entity found at different levels of granularity (as they exist in structured knowledge resources)?*; 2) *What type of attributes are easier to predict, which attributes do models find challenging and, what are the factors that potentially influence these predictions?*

To the best of our knowledge, these aspects have not been systematically addressed previously and our pilot study aims to explore these questions by the way of experiments and analyses that we describe in this chapter.

Plan of the Chapter: In Section 5.2, we begin with a discussion on previous work related to structured knowledge prediction as a task, the datasets used in such tasks and other works with which we identify parallels within our work. In Section 5.3, we discuss what one might expect in terms of extractable information from distributional representations and how can it be best extracted. In Section 5.4, we describe the experimental setup, i.e. the datasets, the models and the evaluation techniques used. Next, we discuss the results of the experiment in Section 5.5 which is followed by detailed analyses of the fine-grained attributes predicted by our model and the factors that potentially affect model prediction capabilities. In Section 5.7, we conclude the chapter by listing our key findings and the limitations within the current setup which motivate further optimization of fine-grained attribute prediction in the next chapter.

TABLE 5.1: Semantic relations between entities and their prediction accuracies (precision).

Relations	TREC-9	CHEM
is a	0.73	0.85
part of	0.80	0.60
succession	0.49	–
reaction	–	0.91
production	–	0.72

5.2 Related Work

KBC Models: There is a large literature on exploiting corpus evidence, through distributional semantic methods, in order to construct and populate structured knowledge bases (KBs) (for example Buitelaar and Cimiano, 2008, and references therein).

Ruiz-Casado et al. (2008) create a knowledge base of entities along with their attributes (representing taxonomic and non-taxonomic relations) by using 1.3 million Wikipedia articles as a textual repository. They extract (an unspecified number of) entities primarily through Named Entity Recognition and Classification techniques (Section 2.5) and use Wikipedia page links as well as pre-determined lexico-syntactic relational patterns to extract 30 entity attributes.

Pantel and Pennacchiotti (2008) use their *Espresso* algorithm to extract entities and their attributes from two datasets: TREC-9 (Voorhees, 2001) and CHEM (Brown et al., 1994), based on lexico-syntactic seed patterns. As depicted in Table 5.1, they define attributes from 5 semantic relations between entities, and report precision for each. *succession* indicates a person succeeding another in a position or title, hence, not relevant for CHEM which contains text from the domain of chemistry. Conversely, *reaction* and *production* are semantic relations specific to chemistry, and therefore, not relevant for TREC-9 which is text collected from news.

This line of work, however, does not attempt to *connect* entity representations extracted from corpora and from knowledge bases, as we do.

Socher et al. (2013a) represent WordNet and Freebase entities with corpus-based distributional vectors. Given an (entity$_1$, relation, entity$_2$) triple, they train a tensor for each relation of interest to return high scores when combined with the vectors of two entities that hold the intended relation. At test time, the system is used to classify relational triples as true or false, as well as to predict new entities that hold a certain relationship with a target entity. However, they consider only 7 relations in total. Freitas et al. (2014) use a hybrid distributional relational semantic model for selective common-sense reasoning by using a DSM as a complimentary semantic layer to a relational model created from a knowledge base. They use ConceptNet (Speer and Havasi, 2013) to extract (entity$_1$, relation, entity$_2$) triples which are finally mapped to a distributional space. They then proceed to identify all permissible paths between a given source and target (both entities) such that all elements in the path are semantically related to their neighbours within a specific threshold. On an evaluation set of 51 (source, target) word-pairs [1], for paths with length (2,3 and 4) they report a score[2] of (0.60, 0.15 and 0.01) on meaningful path selection and an accuracy of (0.95, 0.82 and 0.73) on identifying the correct paths based on semantic relatedness. Freitas and Curry (2014), on similar lines as above, propose a natural language interface that uses a compositional distributional semantic framework which is mapped to a linked-data model created from a knowledge base. They evaluate their system on an open domain dataset (QALD 2011) to achieve a mean average precision of 0.62 and an average recall of 0.81 on 102 natural language queries.

A more recent line of work follows the objective of addressing KBC by constructing optimized embeddings for link prediction. For example,

[1] Related under the context of the Question Answering over Linked Data challenge (QALD 2011/2012) `http://www.sc.cit-ec.uni-bielefeld.de/qald-1`

[2] Computed as the ratio between *the number of paths selected using their reasoning algorithm* by *the total number of paths for each path length.*

Nickel et al. (2016) use distributional representations to generate optimized Holographic embeddings to generate compositional embeddings of binary relational data and, Trouillon et al. (2017) demonstrate that complex embeddings are more optimized as compared to real valued embeddings for link prediction. This is because complex embeddings lead to better model performance in both symmetric and asymmetric relation prediction, whereas, real-valued vectors to not work well in the latter case. Additionally, complex vectors are supposed to be more computationally efficient as compared to the real-valued vectors while being equivalent to holographic embeddings (Trouillon and Nickel, 2017).

Kotnis and García-Durán (2019) predict structured knowledge, specifically numerical attributes, from customised datasets created from existing state-of-the-art datasets for KBC (described below). Their prediction models accept knowledge-graph embeddings, which in principal, are very similar to the construction methodology followed by Bordes et al. (2013) (and other offshoots); the key difference being that their embedding learning process includes not just the information from the knowledge graphs but also artificially constructed negative samples by randomly (but uniformly) replacing the target entities in the entity triples. Their embeddings are also injected with numerical attribute information by adding weighted attribute values to the learning objective. They evaluate numerical attribute prediction using both regression modelling and label propagation (for numeric attributes) in a graph (Fujiwara and Irie, 2014). Our similarity with Kotnis and García-Durán (2019), in terms of learning, only exists in the prediction of numerical attributes through regression. While our embeddings come from a comparatively generic source of information i.e., text, their embeddings are constructed from a knowledge base. Additionally, they report that numerical attributes are difficult to learn via regression and estimation of numeric attributes from their nearest neighbours is seemingly a more effective strategy. Basically, numerical attributes cannot be easily learnt through a large collection of (plausibly unrelated or distantly related) attributes. We, on the other hand, show that it is indeed possible to extract

TABLE 5.2: Popular KBC datasets and their statistics.

Dataset	#Relations	#Entities
WN11	11	38,696
WN18	18	40,943
FB13	13	75,043
FB15K	1,345	14,951
FB15K-237	237	14,541

such information from seemingly (and distantly) connected attributes.

The above and other related or similar works, focus on harvesting entities, a limited number of discrete attributes (i.e., relations between entities) or exploiting distributional semantics to navigate among them. We on the other hand focus on predicting full-fledged attribute-based descriptions of entities – as seen in knowledge bases.

KBC Datasets: In the distributional KBC space: WN11 and FB13 by Socher et al. (2013a), WN18 and FB15K by Bordes et al. (2013) and FB15K-237 by Toutanova et al. (2015) are the most widely used benchmark datasets for model evaluation. We list these databases in Table 5.2 along with the total number of unique semantic relations that they contain as well as the total unique entities that hold these semantic relations. The WN datasets (the first two in the table) are WordNet-based datasets whereas the FB datasets (the latter three) are extracted from Freebase. The FB15K dataset presents a shortcoming in the form of redundant datapoints that reflect the same relationship in a reversed order. To mitigate this problem, a subset of this dataset was created and named FB15K-237 that removes the inverse-duplicate relations and consequently results in a dataset with a reduced number of unique relation types (from 1,345 to 237). Since the pruned relations are redundant, the number of entities in FB15K-237 does not reduce significantly.

We deliberately forgo using these datasets (and their subsets or derivations) for our experiments because while they might serve the purpose

of validating a part our primary hypothesis (that fine-grained information is encoded in distributional representations), they deviate from the focus we put on entities with respect to a comprehensive analysis of their range of extractable attributes, their granularity and limitations of distributional models in predicting this range. Moreover, as it will become evident from the discussion in Section 5.4.1, our datasets by their design satisfy all the questions we have raised while not only containing a much higher number number of unique relation types but also a (comparatively) diverse set of relations.

Cognitive and other non-KBC Models: There is also considerable work on mapping between corpus-based word representations and other representational spaces, such as subject-generated concept properties (Johns and Jones, 2012; Hill et al., 2014; Făgărăşan et al., 2015), visual features (Frome et al., 2013; Socher et al., 2013b; Lazaridou et al., 2014) or brain signals (Mitchell et al., 2008; Murphy et al., 2012b). While we share our methodology with these works, in all these settings the focus is entirely on predicting numerical attributes, whereas we treat both numerical and binary attributes. Rubinstein et al. (2015) use distributional vectors to predict binary conceptual attributes of common nouns, as well as a continuous score measuring saliency of such attributes. However, the fine-grained attributes that we intend to study are conceptually very different from those of all these studies.

Weston et al. (2013) use statistical methods to embed words tokens and knowledge base entities and relationships in a vector space for optimized relation extraction. Our proposal is only distantly related to these methods since this line of work does not use distributional semantics to induce word vectors, and ignores numerical attributes.

Herbelot and Baroni (2017) use Wikipedia definitions of entities to predict unseen entity representations with a Mean Reciprocal Rank of 0.04 on 300 test datapoints. A quick look by the reader at random Wikipedia pages (for example, a list of countries) would make it evident that these

definitions are loaded with fine-grained information – mostly spatial descriptions that are largely categorical (and sometimes numeric as well). We interpret their results as evidence in support of distributional information being successfully modelled by DSMs from lexical concepts that represent referential information.

Herbelot (2015) construct vectors for individual entities (literary characters) by contextualizing generic noun vectors with distributional properties of those entities; a more detailed description can be found in Section 3.2.1.1. At a broad level, we share our goal of getting at referential information with distributional semantics, however, we contrastively work with real-world entities and fine-grained information.

To sum up, prior work that models structured knowledge is predominantly focussed on task-specific optimization, with less focus on empirically analyzing the factors on the linguistic side. On the other hand, the distributional semantics community too has paid less attention on making a concrete connection between lexical and referential information.

5.3 Assessing Encoded Knowledge in Vectors

One may not have looked up the entry for Denmark in an encyclopaedia, but, based on the general impression of this country from the news, from movies, etc., a person can be pretty confident that it is a representative democracy, that its GDP per capita must be close to that of Sweden, and so on.

With this experiment we try to accomplish the same feat in a distributional setup. We hypothesize that there is a systematic correlation between sets of linguistic contexts, encoded in the distributional representations of entities and their corresponding fine-grained attributes – as seen in knowledge bases. Consequently, the most pertinent question, and also an old one in distributional semantics, that arises is: *what* does it mean for information to be 'encoded' (or more generally, included) in distributional representations? Due to the uninterpretability of the vectors (see Section 1.1.2), the

only way to assess the information in a vector is through drawing semantic inferences via supervised or unsupervised learning. Between the two, supervised learning provides the advantage of choosing what we want to extract. In other words, we have the choice to select the semantic properties that we want a model to make estimations upon as well as to empirically assess those estimations against what one believes are the expected outcomes. If the (correct) estimations can be generalized then we can conclusively say that the information is indeed 'encoded' in the vectors.

Note that similar conclusions can also be made by using unsupervised approaches, like clustering. However, as described in Section 2.3, such techniques often require further human analysis with their results being open to interpretation. Moreover, the results are typically salient patterns within the data and their correlations (similar to the clustering results we observed in Section 3.4.2.2); whereas our focus is on investigating for specific patterns of information. Ultimately, what we want to test through a supervised setup is not just whether information can be extracted from vectors but whether it is extractable from the combination of vectors and the labels of the training set that we use to train our models.

Within the scope of this experiment, we intend to use distributional representations of entities for zero-shot learning of their corresponding fine-grained Freebase attribute-based representations. It is zero-shot learning in the sense of Palatucci et al. (2009): we split the datasets at the entity, rather than attribute level, such that at test time our system must predict the full attribute set of entities that were not seen during training at all. The attributes are categorical as well as numeric Freebase attributes (described in greater detail in Section 5.4.1.1), and our model does not distinguish between them. For categorical attributes, we interpret the value returned by the model as the probability of "success" of a binary Bernoulli trial. In the numeric case, we view the probability returned by the model as directly representing normalized attribute values.

In terms of general approach, the aim of this chapter (and the next one as well) is not to optimize Knowledge Base Completion as an NLP task but

to use its framework as a test-bed to explore the bounds of distributional semantic models in terms of the information they capture.

5.4 Experimental Setup

As is evident from the discussion in Section 5.2, prior work in distributional semantics as well as in knowledge representation areas have not empirically investigated the systematic relations between distributional features and fine-grained attributes of entities.

Through this pilot study we aim to discover such systematic relations. In this experiment, within the bounds of a supervised setup, we will identify the limitations of distributional semantics in terms of fine-grained attribute prediction.

The next subsections are devoted to describing the experimental setup: we first describe the datasets that we use for the experiment, followed by a definition of our model, baseline and the upper bound as well as the evaluation scheme used.

Thereafter, we will discuss our results and support them with a qualitative analysis in the succeeding sections.

5.4.1 Data

As mentioned in Section 5.2 (see KBC **Datasets**), we create our own datasets. Conforming to the agenda of the experiment, the datasets contain distributional as well as fine-grained attribute representations of entities. The distributional representations come from the semantic space described in Section 3.3.1, where each entity is represented by a 1000 dimensional vector constructed from the 100 billion word Google News corpus. The fine-grained attribute representations are constructed from Freebase. We discuss their construction methodology in the next subsection, and then we describe the dataset design for the experiment.

Attribute	Value
geolocation::latitude	52.52
geolocation::longitude	13.38
fertility_rate::1960	2.37
fertility_rate::1994	1.24
fertility_rate::2010	1.39
date_founded	1871-01-18
containedBy	Western Europe
containedBy	Europe
containedBy	Eurasia
adjectival_form	German
member_of::organization	world_bank

TABLE 5.3: Sample of numeric and categorical Freebase attributes for *Germany*.

5.4.1.1 Attribute-based Entity Representations

For each *entity*, Freebase contains a list of (*attribute* - *value*) tuples, where values can in turn be entities. Table 5.3 shows a sample of the attributes recorded in Freebase for the country *Germany*. Note that some attributes are simple, like `date_founded` – that can be reached in the Freebase graph in 1 hop, while other can be called complex, in the sense that they are attributes of attributes (e.g., `geolocation::latitude` – 2 hops in the Freebase graph). We use a double-colon notation to refer to complex attributes. The values of all attributes can be either *numeric* or *categorical*. The numeric attributes in particular are often strongly correlated, both within attributes types across years (e.g., fertility rate in different years) and across attributes within years (e.g., absolute GDP and GDP per capita in a given year).

To create the attribute-based representations for entities, we first record all simple attributes as well as complex attributes of at most two hops in the Freebase graph, without manual inspection. We limit ourselves to two hops purely out of semantic considerations – that the 'core meaning' of an entity is constituted by its attribute-values pairs with one or two hops. The greater

the number of hops, the more complex the attribute-value becomes i.e., the number of semantic relations between the two increases. Intuitively, one can expect a high correlation between the number of semantic relations and the semantic distance. As a consequence, we assume that as the semantic distance between the attribute and its value increases, the chances of that (attribute – value) being encoded in the distributional representation of that entity (by the way of co-occurrence based contextual information) decreases.

Next, we linearly rescale all numeric attributes to $[0..1]$. The rescaling provides a lower and an upper-bound to the attributes due to which the supervised learning algorithm is not adversely affected by acute (weighted) values during training. Note that for evaluation, we transform the attribute values back to their originally observed range. We also translate all categorical attributes into a binary representation by suffixing the original value to the original attribute name. For example, the attribute `member_of::organization` with the value `world_bank` results in a binary attribute `member_of::organization::world_bank` having the value 1 for all (and only) those countries that are members of the World Bank, 0 for the others[3].

While the approach we have adopted for categorical attribute learning is simple and well suited for linear models, it has certain limitations: 1) it results in a very high number of fine-grained categorical attribute-value pairs per entity which in turn leads to high data sparsity; and, 2) it reduces the ability of the model to generalize over unseen categorical attributes to a certain extent.

On the grounds of this experiment being more of a proof-of-concept rather than an optimization task, we mitigate the aforementioned issues by applying a threshold on the attribute frequency for an attribute to be considered as a dimension of an entity (discussed in the next subsection).

[3]We considered treating some categorical attributes as multi-valued, but decided against it since the cases in which alternative values are mutually exclusive are rare (e.g., the same country can be `containedBy` multiple entities, see Table 5.3).

However, in Experiment 2 (Chapter 6), we will present an effective optimization on categorical attribute learning.

5.4.1.2 The Two Datasets: Countries and Cities

Domains: Since our pilot study aims to investigate the fine-grained information in distributional representations, we want to focus on those sets of entities which we expect to be most diverse and populated in terms of their fine-grained attributes in the knowledge base. This would enable us to not just address our primary hypothesis but also provide answers to the ancillary questions that we raise at the end of Section 5.1. Note that the other popularly used datasets (Section 5.2) are not domain specific. Another factor behind the choice of domains is the ease of verifiability of correctness of predictions, and therefore the soundness of our approach, not just by technical experts but also by ordinary persons.

To this purpose, we identify two domains (also called *types*) in Freebase whose entities are constantly curated in terms of adding new or updating old information: *countries* and *cities*. We consider two different domains in order to check that the mapping that we seek can be established for more than one type of entities. In Experiment 2 (Chapter 6), we study a much larger variety of domains, such as people, animals and organizations; with specific focus on categorical attributes.

Datasets: We build two datasets, one for countries and the other for cities, with data automatically extracted from Freebase[4].

The *Countries* dataset consists of the 260 countries. Some countries are strictly historical, like Yugoslavia, but, since this does not impact our method, we keep them in the dataset. All entities in this dataset have a mapping to our distributional space as well as Freebase. Therefore, the distributional representations come from our distributional space and, the fine-grained attribute representations from Freebase (as described in

[4]Both datasets are publicly available at `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/CityCountry.html`.

Section 5.4.1.1). Attributes that occur less than 15 times are discarded, since they are either not consistently recorded or rare. This results in a total of 707 numeric and 247 binary attributes. Finally, we partition the data into training, validation and test sets (60, 20, 20%) with 156, 52 and 52 countries respectively.

We apply the same process to the *Cities* dataset. We select 1645 cities from the intersection of the distributional space and the Freebase city list. We populate the datasets with corresponding distributional representations and Freebase representations. In this case, we have 211 numeric and 106 binary attributes – the numbers are smaller because countries have a richer representation in Freebase than cities. With the same partitioning constraints we get 987, 329 and 329 cities in the training, validation and test sets respectively.

5.4.2 Model, Baseline and Upper-bound

Model: Our main model, called *Dist2Ref*, is designed using logistic regression which is easy to implement, interpret, and very efficient to train. More concretely, we train a logistic regression model (essentially, a linear function) that, given a distributional representation for an entity, predicts what its Freebase attribute values are. In our approach, each Freebase attribute is predicted with an independent logistic regression model based on a constant set of input features (1000 distributional dimensions).

Logistic regression, by its design is apt for binary attributes but, recall that our space also consists of real-valued numeric attributes and, we plan to model them jointly in this experiment. Therefore, to introduce non-linearity we use the sigmoid function as the activation function. Consequently, for binary attributes we employ a decision boundary of 0.5. We optimize the parameters with gradient descent, using the Cross Entropy loss function, as defined in the equation below; where y_n are the true predictions, $\hat{y}_n$ are the model estimations and N is the total number of samples.

$$-\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right] \tag{5.1}$$

It is important to note that the gradient of the cross-entropy loss for logistic regression is the same as the gradient of the squared-error loss for linear regression (Murphy, 2012). Therefore, we could have also designed this joint prediction task using linear regression with a squared-error loss function where we round-off the predictions for the binary attributes. However, cross-entropy loss has a slight advantage over squared-error loss as it penalizes confident and wrong predictions more than it rewards confident and right predictions. Due to this, we proceed with logistic regression.

We considered L_2 regularization to address possible overfitting, but experiments on the validation set showed that the model performs best without any regularization. *Dist2Ref* does *not* take advantage of the correlations between the output attributes mentioned in Section 5.4.1.1.

Finally, the model selection (*Dist2Ref*) is done on the *Countries* dataset. The same model is then subsequently used to train and test on the *Cities* dataset as well.

Baseline: Our baseline is constructed from the attribute values in the training set. For binary attributes, we predict the majority class (0 or 1) and, for numeric attributes, we predict the mean value of the attribute.

Upper-Bound: We also train an upper-bound, called *Ref2Ref*, that uses the same architecture as described above but uses as input not distributional vectors but the Freebase attributes-based representations themselves (as described in Section 5.4.1.1). In other words, this model has to learn "only" an identity mapping.

This is not trivial, though, due to the presence of strong correlations among attributes, in particular the time series attributes (see `fertility rate` in Table 5.3). This is because regression models tend to estimate the relationship between each independent variable in the input and the

dependent variables in the output, independently; in our case these variables are Freebase attributes. While it is always a possibility that the input and output attributes indicates correlation i.e., their relationship indicates correlation, a strong correlation between a set of input attributes reduces the robustness and reliability of the model. Strongly correlated input attributes result in model parameters being highly susceptible to small changes, brought by their addition, deletion or alteration during training. One major consequence is that significant attributes might become insignificant (and vice-versa), thus, preventing the model from capturing the most influential attributes. In other words, the model is unable to learn the (correct) semantic patterns that it is expected to learn. For a more detailed discussion, see Farrar and Glauber (1967) and Daoud (2017).

5.4.3 Evaluation

Since there is no appropriate unified evaluation measure that covers both numeric and binary attributes, we evaluate them separately.

The binary attributes, as mentioned in Section 5.4.1.1, have a value of 1 or 0 indicating the truth value of whether an attribute is present in an entity or not, respectively. Since the outcome is discrete, therefore, we report the mean accuracy.

The numeric attributes, on the other hand, represent continuous values. Therefore, for them, we consider attribute prediction as a ranking task. As an example, take the `population::2011::number` attribute, and imagine that we only have three countries (Germany: 80M; Spain: 36M; and, Netherlands: 17M). If we predict 56M for Spain's population, it is still (correctly) predicted as the second most populous country (rank difference of 0); a prediction of 16M, however, would push Spain to third place (rank difference of 1).

This suggests the use of rank correlation coefficients like Spearman's ρ. However, we want to measure not only how well the model can rank the countries in the test set, but also whether these predictions are consistent

with the training set (which makes evaluation both more challenging and more realistic). One way of achieving this goal would be to use ρ on the union of training and test instances, but this could lead to misleadingly high correlation coefficients since this method would include the labels of the training instances in the evaluation.

Consequently, we define our own evaluation measure, following a rationale similar to Frome's evaluation (Frome et al., 2013) of a zero-shot learning scenario. What we evaluate, for each attribute, is the rank of the test countries in the whole country list. Note that this makes our task harder, as there are more confounders: if we only evaluated on the test set, there would be shorter lists and therefore less chances of getting bad rankings.

So, concretely, we first define the prediction quality of each attribute, $Q(a)$ in Equation 5.2, as the median of the rank difference between the prediction and the gold standard in a list that includes both training and test countries (we use the median to give less weight to outlier countries). We also normalize the rank difference to obtain a number between 0 and 1. In the second step, as shown in Equation 5.3, we define the quality of the complete model, the *normalized rank score (NRS)*, as the mean of all attribute quality scores, in parallel to our evaluation on binary attributes.

$$Q(a) = \frac{1}{||I||} \, med\{\mid r(p_a(i), I) \; - \; r(g_a(i), I) \mid -1 \mid i \in Ts\} \tag{5.2}$$

$$NRS = \frac{1}{||A||} \sum_{a \in A} Q(a) \tag{5.3}$$

In Equation 5.2, let the set of instances I be partitioned into training instances Tr and test instances Ts. Let $a \in A$ denote an attribute. We write $p_a(i)$ for the predicted value of attribute a for instance i and $g_a(i)$ for the gold standard value. Finally, let $r(v, S)$ denote the rank of value v in the list resulting when ordering the set S. Subtracting 1 in Equation (5.2) ensures that, when the predicted and gold value of an attribute are adjacent in the

ranking, their rank difference is 0, capturing the intuition of rank difference as counting the number of falsely intervening items.

Note that, when evaluating each instance i, we use gold-standard values for all other instances, so that there the baseline is not hampered by ties.

The NRS in Equation 5.3, can be interpreted as follows: within the range of $[0..1]$, smaller numbers indicate a better ranking, for example, NRS of 0.1 means that the prediction is about 10% of the ranks off (e.g., by four countries in a forty-country list).

5.5 Overall Results

In this section, we show that our model makes predictions that are significantly better than an informed baseline; although they fall short of the upper-bound we have established.

Attribute Type	Model	Countries	Cities
Binary (Acc)	Baseline (most frequent class)	0.86	0.97
	Dist2Ref	0.90	0.99
	Ref2Ref (upper bound)	0.96	1.00
Numeric (NRS)	Baseline (mean value)	0.35	0.35
	Dist2Ref	0.22	0.25
	Ref2Ref (upper bound)	0.14	0.21

TABLE 5.4: Baseline and model predictions for binary and numeric Freebase attributes of *Countries* and *Cities*. Higher Accuracy and lower NRS are better.

Table 5.4 shows the results for predicting Freebase attributes from distributional vectors on the test sets of our two datasets: *Countries* and *Cities*. The table is divided in two halves. We first report the accuracies (Acc) on the Binary attributes, followed by the normalized rank score (NRS) of the numeric attributes. Recall from Section 5.4.3 that both evaluation measures range between 0 and 1. For binary attributes, an accuracy score of 1 is the best. For numeric attributes, an NRS score of 0 is the best. For each

attribute type, the evaluation scores are listed for the Baseline, *Dist2Ref* (our model) and *Ref2Ref* (upper-bound).

The baseline is relatively high, in particular for the binary attributes, many of which are positive for a small subset of entities only. There is a considerable variance between the accuracies of the two datasets, though. For *Countries*, the baseline yields a mean accuracy of 0.86, but it achieves 0.97 on *Cities*. The increase stems from very sparse categorical city features such as `containedBy`, which includes all levels of administrative divisions – that is, for the US, all counties appear as values and are transformed into sparse binary features (see Section 5.4.1.1). On numeric features, where the baseline predicts the mean, its performance is 0.35 NRS on both datasets. In other words, its average prediction is off by about one third the length of the ranked list for each attribute.

Recall that the upper bound model, *Ref2Ref*, uses Freebase attributes to predict Freebase attributes. All it has to learn is that there is one feature in the input that corresponds ideally to the output. This works almost perfectly for binary attributes, with accuracy values of 0.96 for *Countries* and 1.00 for *Cities*. However, its performance on numeric features (with NRS at 0.14 and 0.21, respectively) is not quite perfect. We attribute this to the presence of correlations between the input attributes (Section 5.4.1.1) and their effect on model learning (Section 5.4.2).

The model whose performance we are actually interested in is *Dist2Ref* – in which we map from distributional information to Freebase attributes. The model performs with remarkable consistency between these two extremes. For the binary attributes, the *Dist2Ref* model is almost centrally situated between the baseline and the upper bound *Ref2Ref* of the two datasets: *Countries* ($0.86 < \mathbf{0.90} < 0.96$) and *Cities* ($0.97 < \mathbf{0.99} < 1.00$) datasets. For the numeric attributes as well we see a similar pattern: the *Dist2Ref* model is higher than the baseline, lower than *Ref2Ref* and centrally situated for the *Countries* ($0.35 > \mathbf{0.22} > 0.14$) dataset; although, it is comparatively further away from the baseline and closer to *Ref2Ref* for the *Cities* ($0.35 > \mathbf{0.25} > 0.21$) dataset. Overall, we see a consistent error

reduction of around 30% over the baseline, with a similar distance to the upper bound.

To rule out that we misinterpret our accuracy-based evaluation for the binary features in the face of a highly skewed class distribution, we also computed precision, recall, and F-Score values on the positive class. The relative patterns match those of the accuracy-based evaluation well (Countries: baseline F=0.13, *Dist2Ref* F=0.51, *Ref2Ref* F=0.77) and indicate that generally precision is higher than recall.

As mentioned in Section 5.4.2, the model selection has been done on the *Countries* dataset and, a significance test with bootstrap re-sampling (Efron and Tibshirani, 1994) showed that all pairwise comparisons (Baseline vs. *Dist2Ref*, *Dist2Ref* vs. *Ref2Ref*) are statistically significant at $p < 0.001$.

Given that the Freebase attributes we predict are fairly fine-grained, we think that these are promising results considering we only use generic distributional information as input and only logistic regression for prediction.

5.6 Qualitative Analysis

We take the overall results just presented to suggest that we are able to learn fine-grained attributes (i.e., referential information) from distributional information to a large extent. In this section, we take a closer look at what kind of information we are able to learn, what is beyond the scope of our model and, what are the differences between the distributional representations of entities and their corresponding fine-grained representations (the ones our model predicts). The analysis points both to the inherent difficulty of correctly retrieving certain classes of attributes, and to some intriguing properties of the conceptual nature of the knowledge encoded in distributional data, that bias their predictions about certain objective attributes of geographic entities. The analysis concerns only the *Countries* test set.

5.6.1 Attribute Groups

Due to the large number of attributes, we sort all individual attributes into *attribute groups* by their base name (i.e., the leftmost component of their name; see Section 5.4.1.1), which offers an accessible level of granularity for inspection. We obtain 34 numeric and 40 binary attribute groups with median sizes of 8.5 and 2 attributes per group, respectively. The median size of the numeric attribute groups is higher due to the inclusion of time-series attributes, for instance, the attribute `fertility_rate` across all countries has 52 time-series instances; although their distribution is highly skewed – for developed countries such statistics have been recorded and released publicly for a much longer number of years.

Table 5.5 and 5.6 shows the attribute groups for both types sorted by quality. For each group, we report the average normalized rank score (NRS) and accuracy, respectively, for both *Dist2Ref* and the baseline (BL) models. In both tables, *#Attributes* reflects the number of attributes in that group and *#Countries* is the median number of countries instantiating each attribute in the dataset (out of a maximum of 260 countries). In case the performance of an attribute is worse than the baseline (i.e., $Dist2Ref < BL$), we indicate it with a super-scripted exclamation mark (**!**).

The analysis suggests that there are two main factors that affect the results:

1. The degree to which an attribute is *contextually supported*, that is, to what extent its values can be identified on the basis of the contextual information that is captured in a distributional model.

   In text, attributes can occur in complex as well as general patterns which cannot be picked up easily by bag-of-words models. As an extreme example of an attribute that is *not* contextually supported, consider the numeric ISO code of a country (`iso_numeric` in Table 5.5), whose values are arbitrary: they do not correspond to facts about the world that are reflected in the way people use language, and so cannot be picked up by the distributional model. For this reason,

TABLE 5.5: Normalized Rank Score (NRS) of the numeric attributes of the *Countries* test set in descending order of performance.

Numeric Attributes (Normalized Rank Score: lower is better)

Attribute Group	*Dist2Ref*	BL	#Attributes	#Countries
geolocation	0.07	0.30	2	250
gdp_nominal_per_capita	0.11	0.27	1	172
gni_per_capita_in_ppp_dollars	0.12	0.28	32	155
co2_emissions_per_capita	0.12	0.25	49	157
fertility_rate	0.12	0.24	52	178
calling_code	0.12	0.27	1	205
internet_users_percent_pop	0.13	0.32	22	184
entry	0.14	0.23	2	140
gni_in_ppp_dollars	0.16	0.31	32	154
broadband_penetration_rate	0.17	0.68	15	23
population_growth_rate	0.19	0.31	52	201
military_expenditure_perc_gdp	0.20	0.27	24	128
gdp_real	0.20	0.34	51	149
life_expectancy	0.20	0.24	52	179
electricity_cons._per_capita	0.22	0.36	50	105
gdp_nominal	0.22	0.34	52	157
energy_use_per_capita	0.23	0.39	51	104
population	0.25	0.42	54	202
places_imported_from	0.26	0.29	2	18
iso_numeric[!]	0.26	0.23	1	220
national_anthem_since	0.27	0.43	1	97
championships_athletes	0.28	0.33	1	18
gdp_growth_rate	0.28	0.41	51	154
government_debt_percent_gdp[!]	0.33	0.19	17	24
casualties[!]	0.39	0.35	1	33
athletic_performances_rank	0.43	0.43	1	34
date_founded[!]	0.46	0.41	1	61
date_dissolved	0.48	0.48	1	21
climate_avg_rainfall[!]	0.50	0.38	1	4
force_deployments	0.53	0.58	2	20
religions_percentage	0.58	0.66	2	14
minimum_wage	0.63	0.82	28	17

Dist2Ref does worse than the baseline. Note that, in a sufficiently large corpus, we might indeed encounter statements like *The numeric ISO code for Spain is 724*. However, since distributional models represent words as aggregated distributions of their contexts and compute semantic similarity from these context distributions, the contexts that they use need to be generic enough to yield meaningful overlap between concepts (e.g., words). As a result, distributional models cannot easily represent knowledge of the form "the value for property Y of word/concept X is Z".

2. General properties of the data that affect Machine Learning, most notably data sparseness (see Section 2.1.2.1) and potentially attribute value distributions as well. For instance, `minimum_wage` (in both tables) is an attribute which we found to be sparse in Freebase with respect to countries – median country frequency of 17 and 20 as a numeric and binary attribute respectively. Moreover, there are certain attributes, like `places_imported_from`, `climate_avg_rainfall` and `religions_percentage` in Table 5.5, that are not maintained in a time series but are subject to constant alterations due to their constantly evolving nature. Interestingly, as seen in the table, such attributes also have skewed distributions.

Attributes that *are* contextually supported include, for instance, those related to socio-economic development. These are the attributes that people talk (and therefore, write) about in context of countries being more or less developed, rich, having one or another kind of laws, and this is captured in the abstractions over textual context that distributional models perform.

Contextual support can be indirect as well and `geolocation` (Table 5.5) is a prime example for this. Consider that geolocation is the intersection of the latitude and longitude of a location. It may be observed in text, for example, *The geolocation of Germany is* $51^\circ\ 09'\ 51.23''\ N, 10^\circ\ 27'\ 14.83''\ E$. But we can safely assume that this information occurs rarely (if at all) and

as mentioned above, distributional models fail to capture such patterns. Then how does *Dist2Ref* performs remarkably well on this attribute? Recall from Section 5.1 that studies have shown humans to make reasonably accurate locational assessments (although biased to a certain degree) based on their knowledge about not just socio- economic-cultural factors but also on weather-based heuristics, like number of seasons, typical temperature, rainfall, humidity, flora-and-fauna, etc. So, while distributional models do not capture contextual support for geolocation, they indeed successfully capture topical information which has ample contextual support in text by the way of lexical concepts that describe them.

Fortunately, we find that many Freebase attributes are contextually supported to a substantial degree, even some seemingly arbitrary ones. An appropriate example is `calling_codes` in Table 5.5, which we predict very well. They turn out to be correlated with geolocations: 2X calling codes are located in Africa, 3X calling codes in Southern and Eastern Europe and 4X calling codes in Western and Northern Europe (for comparison, ISO codes are assigned in a roughly alphabetical order).

5.6.1.1 Numeric Attributes

Our best numeric attributes, according to the results reported in Table 5.5, belong to the `geolocation` group (latitude and longitude). We provide a more detailed analysis of these attributes below (Section 5.6.2).

As mentioned above, we excel at many attributes which can be broadly related to the economic and social development of a country, such as GDP, GNI, CO_2 emissions, internet usage (each per capita), or fertility rate. These attributes can be expected to be contextually grounded. For some attributes, like GDP, the contextual grounding can be assumed to be relatively obvious, for example, *Luxembourg* will occur with contexts like "financial hub" or "rich" more than *India* does. However, this contextual grounding can also be surprisingly subtle: for instance, the fertility rate is a function of both general development status (lower rates in more developed

countries) and of specific social factors (higher rates in countries with more support for families, such as *France* and *Finland* compared countries with less support, such as *Germany* or *Italy*). In other words, an attribute like `fertility_rate` may not have much contextual grounding but it can be a function of other attributes (GDP, internet usage, etc.) which are plausibly well grounded contextually.

Around the middle of the table, we find the absolute versions of the developmental cluster above (GNI in $, real and nominal GDP). Evidently, the absolute versions of these attributes are substantially less contextually supported than the relative versions. This is not surprising: while India and China have high absolute GDPs because they are large countries, and for instance Luxembourg has a much smaller one, these numbers are not indicative of the actual conditions in these countries, and therefore also not so clearly correlated with what people write about them. This provides another interesting angle on the difference between distributional and formal knowledge representation. In a formal system, absolute GDP, relative GDP, and population stand in a fixed linear relationship and knowing any two of the three uniquely determines the third – thus, all three attributes have equal status. In our distributional space, their status is clearly different, determined by the conceptual relevance of the different attributes.

Towards the end of Table 5.5, we find more attributes related to socio-economic development, such as `government_percent_debt` and `minimum_wage`. While these should be contextually supported too, the problem here is factor (2) mentioned above, namely severe data sparsity (see column #Countries in the table, which lists the median number of datapoints that exhibit each attribute group). The same goes for the remaining attribute groups, for instance `casualties` (describing the total number of military casualties incurred in history), `date_founded` and `date_dissolved`[5] or `climate_avg_rainfall`.

[5]Note that date-based attributes can be contextually supported: We do better on `national_anthem_since`, for which we have more datapoints (97).

TABLE 5.6: Accuracy of the binary attributes of the *Countries* test set in descending order of performance.

Binary Attributes (Accuracy: higher is better)

Attribute Group	*Dist2Ref*	BL	#Attributes	#Countries
continent	0.98	0.84	4	45
time_zones	0.98	0.93	2	26
containedBy	0.98	0.81	9	49
casualties!	0.96	0.97	2	17
places_exported_to!	0.96	0.98	2	17
member_of	0.95	0.86	25	27
championships_athletes!	0.94	0.96	1	22
military_conflicts	0.94	0.94	2	18
organizations	0.94	0.93	8	20
entry	0.94	0.81	5	30
minimum_wage	0.93	0.93	2	20
gdp_nominal	0.92	0.85	1	213
religions	0.92	0.93	3	23
tournaments_participated_in	0.91	0.91	2	27
places_imported_from	0.91	0.91	2	18
athletic_performances	0.91	0.89	30	26
medals_won	0.91	0.89	29	31
gdp_nominal_per_capita	0.90	0.85	1	215
currency_used	0.89	0.89	2	26
official_language	0.89	0.81	4	32
administrative_area_type	0.89	0.69	1	185
companies_founded	0.89	0.83	3	39
organizations_founded	0.89	0.83	3	39
schools_founded	0.89	0.83	3	39
olympics_participated_in	0.88	0.81	9	55
tour_operators	0.88	0.89	3	40
athletes	0.88	0.86	48	36
languages_spoken	0.88	0.84	5	38
government_bodies	0.88	0.87	2	34
administrative_parent	0.87	0.69	1	185
gdp_real	0.87	0.73	1	189
gni_in_ppp_dollars	0.87	0.62	1	170
gni_per_capita_in_ppp_dollars	0.87	0.62	1	170
is_clear	0.87	0.87	1	23
governing_officials	0.86	0.82	14	34
form_of_government	0.84	0.81	11	42
equivalent_instances	0.79	0.75	1	200
exceptions	0.69	0.67	1	87
loc_type	0.69	0.58	1	146
adjectival_form!	0.65	0.69	1	65

5.6.1.2 Binary Attributes

The binary attributes, in Table 5.6, show a similar picture. We obtain good results on meaningful attributes that are arguably strongly contextually grounded, such as geographical and geopolitical attributes (`member_of`: membership in international organizations; location on a `continent`, etc.). However, we fare relatively badly on government-related attributes, like `form_of_government` and `governing_officials`. While this seems surprising at first glance, the attribute `form_of_government` in Freebase makes very fine-grained distinctions: its values include "unitary state", "presidential system", "parliamentary system" and "republic", which are not mutually exclusive, and misses obvious alternatives like "authoritarian system". It is not surprising that distributional models cannot make such subtle distinction between presidential and parliamentary systems. The attribute `governing_official` presents a similar case. Other bad attributes are very domain-specific, including `athletes`, encoding the athletic disciplines that countries participate in (such as swimming, judo, running, etc.), and the data sparsity issue is certainly worse for the binary attributes.

5.6.2 Case Study: Geolocation

We now analyse `geolocation`, our best attribute group, to assess the veracity of our postulations that there is a strong correlation between the ground truth and the information captured by distributional models and, that distributional models also capture conceptual biases. These biases become evident when we analyse the estimations made by models that use distributional representations.

Correlation between physical and estimated distances: In Section 5.1, we discussed how Louwerse and Zwaan (2009) showed that geometric distances, estimated from distributional spaces, have a high correlation with actual physical distances between locations in the real world. In

Model	Countries	Cities
DistRep	-0.36	-0.45
Dist2Ref	0.49	0.88

TABLE 5.7: Pearson correlation coefficients of actual vs. model-predicted distances between countries and cities. All results are highly significant: $p<10^{-14}$.

table 5.7, we report the correlation between real and model-predicted distances for countries and cities. First, we compute the pair-wise *great circle distances* (Kern and Bland, 1948) between items from the *Countries* as well as the *Cities* dataset using their longitudes and latitudes from Freebase. Second, for *DistRep*, we compute the cosines between the corresponding distributional representations of the location pairs and, for *Dist2Ref* we compute the distance by using the latitude and longitude values predicted by the model. Finally, we compute the correlation between the Freebase-based *great circle distance* and the *DistRep* cosines as well as the *Dist2Ref*-based distances.

For countries, as shown in Table 5.7, the correlation is -0.36 for *DistRep* (negative, because cosine is a *similarity* measure), 0.49 for *Dist2Ref*. For cities, *DistRep* reaches -0.45 correlation, and *Dist2Ref* distances are at 0.88, showing that the method can estimate city positions to a perhaps unexpectedly high degree of accuracy[6]. Overall, the results show that the distance information estimated by the *Dist2Ref* model is even more precise than the distance information from distributional representations – *DistRep*; all the correlations obtained are highly significant ($p<10^{-14}$).

The results suggest that we manage to objectify the information in the distributional model, anchoring the entities more firmly in the external world. In Figure 5.1, we validate this statement by marking the actual geolocations (latitudes and longitudes from Freebase) of a random sample

[6]The results are confirmed when the analysis is repeated using the Spearman correlation measure: The *Dist2Ref* coefficients are stable, whereas those of *DistRep* go down to 0.22 (countries) and 0.40 (cities), respectively. The good results for Spearman, as a rank-based measure, indicate that our success is not dominated by outliers.

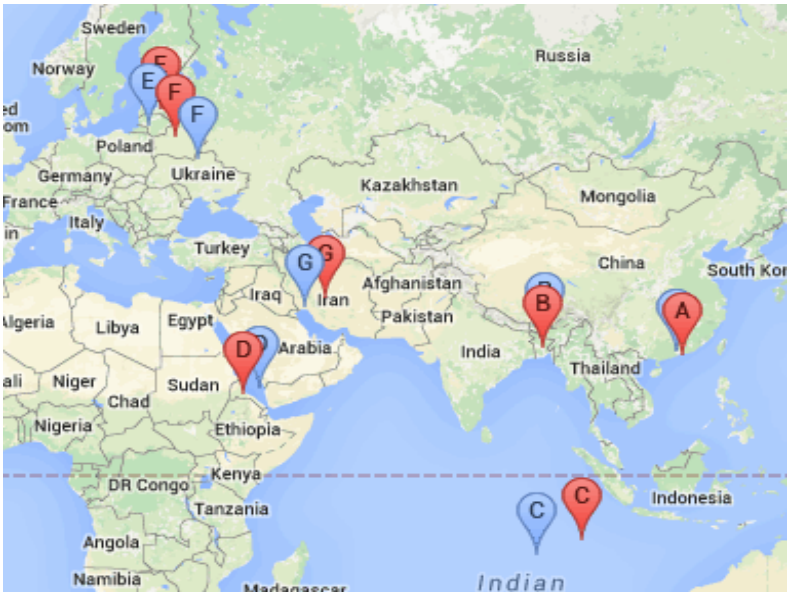


FIGURE 5.1: Geolocation: top model-predictions (in blue) vs. the actual geolocations (in red).

from *Countries* along with their predicted locations by *Dist2Ref*. There are 7 locations in all: (A – Hong Kong), (B – Bangladesh), (C – Cocos Islands), (D – Eritrea), (E – Latvia), (F – Belarus) and (G – Iran).

The figure shows that while there is no consistency between the relative positioning (left-right-top-bottom) of the actual and predicted geolocations, most of the predictions lie in the same country or at least quite close to the actual co-ordinates; with the exception of Cocos Islands (C) and Eritrea (D). Both the countries are of a relatively small size, not as prominent geo-politically as compared to most of their neighbours and hence, have a rather infrequent coverage in global news. Consequently, one would expect their distributional vectors to be informationally deficient in terms of providing contextual support to their Freebase attributes. Nonetheless, their predicted geolocation too is not that far off from the actual location if one views the differences on a global scale.

FIGURE 5.2: Conceptual biases through Geolocations: model-predictions (in blue) vs. the actual geolocations (in red).

Conceptual Biases: As mentioned in Section 5.1, both humans and estimations made from conceptual spaces are found to suffer from biases resulting from cognitive, socio-economic and cultural factors. We observed systematic conceptual biases in our distributional representations as well. For example, in *DistRep* the countries whose distance is overestimated the most almost invariably lie across the Europe / Maghreb + Middle-East divide: (*Slovakia – Syria*), (*Spain – Libia*), etc. Among the country pairs that are close in distributional space but geographically far, we report high similarity in pairs like, (*England – New Zealand*) due to both being insular, rainy and English-speaking, (*Bahamas – Seychelles*) plausibly due to both being beautiful archipelagoes and (*Brazil – China*), the recent economic powerhouses.

Interestingly, *Dist2Ref* does also show some cultural effects in its geolocation errors, as seen in Figure 5.2. The figure shows a set of 7 islands: (A – New Caledonia), (B – Cocos Islands), (C – Cook Islands), (D – Mauritius), (E – Niue), (F – Tuvalu) and (G – Vanuatu). Except for Cocos Islands (B) and Mauritius (D), all are located in the south-pacific ocean but have been predicted next to Madagascar. One plausible

explanation is that these lesser-known islands states have been placed close to the well-known prototype for beautiful island. However, look at Mauritius (D), which is already close to Madagascar but has been predicted in India. This is a strong indication of cultural bias due to Mauritius historically being a British colony, colonised by Indians. In that light, it is plausible that the other locations in pacific (having a strong French connection) have shifted towards their larger cultural counterparts; due to being predominantly colonised by French (i.e., most of the eastern African subcontinent, including Madagascar). We also observe similar patterns in Central American countries (such as Panama, El Salvador, and Nicaragua) move towards their "cultural center of gravity", South America.

In line with our goal to extract fine-grained / referential attributes, thus, we are satisfied. Despite conceptual biases, our simple model *Dist2Ref* is able to distill the referential part from the distributional representations.

5.7 Conclusion

With this experiment, we have shown that a simple model can learn to predict, to a reasonable degree of accuracy, fine- grained (referential) attributes of an entity that are typically seen in a knowledge base from the corresponding corpus-based distributional representation. When evaluated on the prediction of both categorical and numeric attributes of countries and cities, the model consistently reduces baseline error by 30%, and is not far from the upper bound. Further analysis suggests that our model is able to "objectify" distributional representations for entities, anchoring them more firmly in the external world in measurable ways.

Overall, the results suggest that, while distributional semantic vectors can be used "as-is" to capture generic word similarity, with some supervision it is also possible to extract other kinds of information from them, including structured factual statements of the sort encoded in manually-curated knowledge bases. This makes distributional vectors very attractive as general-purpose word meaning representations.

We have also shown that some of the errors in the predictions can be explained on cultural grounds, but that these effects are more pronounced in the input of our model, a standard distributional semantic model, than in its output. Our analyses also suggest that the main limiting factor in learning referential attributes, apart from good old data sparseness, is the degree to which they are *contextually supported*, that is, to what extent they are expressed with consistent and specific linguistic means in the context of their target words. This determines whether they are actually represented in the distributional model in the first place.

One limitation of this work is the treatment of categorical attributes, which are essentially multi-valued attributes, as binary attributes. This restricts the model from predicting binary attributes at test time which have not been observed at model training. Another shortcoming is the prediction of fine- grained information on two closely related domains: countries and cities.

We address these limitations in Experiment 2 (next chapter) where we optimize categorical attribute prediction by using distributional representations, instead of binary values, as the model output. We also experiment with a much larger set of domains that are diverse, both in terms of data frequency and entity types.

Chapter 6

Fine-grained Attribute Prediction - Experiment II

In Chapter 5, we conducted a pilot study (Experiment I) that predicted numeric and categorical fine-grained attributes of entities, in a structured format (as seen in knowledge bases), from their distributional representations. Our model performed reasonably well on both types of attributes. However, since the nature of the study was predominantly exploratory, our focus was on producing a proof-of-concept with the simplest possible method. Consequently, while numeric attributes were treated effectively, we observed certain limitations related to categorical attribute prediction (Section 5.4.1.1) – the main problem being the inability of the model to predict unseen values. We alleviated this in a subsequent experiment.

In this chapter, we describe this follow-up experiment to our pilot study which specifically deals with distributed prediction of categorical attributes. At the same time, we go beyond the subset of entities from 2 domains (*Countries, Cities*) in the previous experiment to exploring a much larger variety of entities from 7 diverse domains. Owing to the data diversity, we are also able empirically analyse if model performance is dependent of factors other than data sparsity; for instance, the presence or lack of contextual support for an attribute in distributional vectors – a phenomenon which we posited in the previous chapter through a qualitative analysis of our results on mainly numeric attributes. Now, our analysis is on categorical attributes, thus, bridging the gap of Experiment I.

6.1 Motivation

Our previous experiment was a success in the sense that it showed our hypothesis, that fine-grained information is encoded in distributional representations, to be correct. Experiment I treated categorical attributes as binary attributes so that a simple logistic regression model can be trained to classify an instance of an attribute as true (1) or false (0). For example, in *Germany* the *official language is Deutsch* is 1 and *Germany* lies in the *continent of Asia* is 0. We reported an accuracy of 0.90 and 0.99 on the *Countries* and *Cities* datasets respectively, which was in-between the established baseline and upper bound. However, from a technical point-of-view, we observed two problems.

The first problem results from the multi-valued nature of categorical attributes. For each entity, the output vector constructed (via Freebase) adds as many dimensions as there are values associated with an attribute. Thus, leading to a very high dimensional sparse entity vector. For example, consider the attribute `president` of a country. At the time of this work, the Freebase knowledge graph contained a list of 44 presidents (uptil Barack Obama) for USA, and on similar lines, each country maintained its own list of presidents (wherever applicable); although the length of the lists varied depending on the detail with which the data for a country was recorded. Our strategy was to pool all the attributes together and then create output entity vectors from this pool, so that the vectors have identical dimensionality. However, the pool size exploded dramatically with attributes like `athletes` which record historical (and present day) data of the personalities associated with sports within a country, across all the countries. In our case, the resultant output vectors originally had a dimensionality of about 60K dimensions (mostly sparse) which posed a significant learning challenge for a simple supervised setup.

We solved the first problem by putting a threshold of atleast 15 occurrences for an attribute-value to be pooled and subsequently considered as a dimension, while keeping only those categorical attribute-values in the pool

that are true (1). This brought down the dimensionality to 954 and 317 for the *Countries* and *Cities* datasets respectively (Section 5.4.1.2). While the output dimensionality became manageable, this brought forward the second problem: By design, the entities in test set are not seen during training, and consequently, the model cannot efficiently learn to predict unseen values. In other words, the model lacks the ability to generalize since it has only been trained on positive examples without any confounders whatsoever.

In order to avoid the problems mentioned above and at the same time deal with the multi-valued aspect of categorical attributes, we now design a system that does not treat categorical attributes as binary attributes. Instead, the system tries to learn the distributional representation of the value of a categorical attribute. This allows the model to generalize more effectively over multi-valued unseen data in the test set.

Learning the distributional representations of categorical values has another advantage. Recall that in Section 5.6.1.2 we observed that the values for certain categorical attributes are not mutually exclusive. For example, for the attribute `form_of_government`, the following value pairs have a considerable semantic overlap: (*republic* and *parliamentary system*) or (*presidential system* and *semi-presidential system*). In such cases, learning a distributional vector might be more beneficial as the models are more likely to discern the finer distinctions between these values via learning their distributional features (instead of learning a *True* or *False* classification).

Note that for the sake of easier understanding, from this point onwards, we will refer to categorical attribute-value pairs as *categorical relations*. The two are essentially the same because a categorical attribute holds a semantic relation between an entity and a categorical value. Consequently, our usage of *learning categorical relations* is synonymous to learning the value of a categorical attribute.

Plan of the Chapter: We start by describing previous work which is most related to (or, that motivates) our work on categorical relation prediction, in

Section 6.2. In Section 6.3, we describe the design of the current experiment where we describe the models and the evaluation scheme used. Based on the design, in Section 6.4, we discuss the categorical relation datasets we have constructed. We then discuss the results of modelling categorical relations in Section 6.5 and, we follow it up with a detailed analysis of the factors that influence categorical relation prediction in a distributional setup in Section 6.6. We then describe how some of the previous state-of-the-art work can be adapted and made comparable to our work in Section 6.7. Finally, in Section 6.8, we conclude Experiment II by highlighting our key insights from our analyses as well as briefly listing out the direction that we further pursued in fine-grained relation prediction.

6.2 Related Work

While there are many approaches that deal with relation prediction for knowledge completion, in this section we discuss (and constrast) only those which are closely related to our model architecture.

Bordes et al. (2013) introduce a system, called *TransE*, that models relationships by viewing them as 'translations' between entities. They assume that in a data triple (*head, relation, tail*), where head and tail are entities, if the relation holds between the entities then, $head + relation \approx tail$, i.e., the tail entity should be close to the combined representation of the head and relation. The goal is to auto-complete knowledge base data, which can be viewed in similar triple-based structures, by identifying the connectivity patterns between entities and then generalizing these learned patterns to relationships between a specific entity and all the others. To model the entities and relations, they use the WN18 and FB15K datasets (as described in 'KBC Datasets' in Section 5.2). On test sets of 5000 and 59,071 datapoints for WN18 and FB15K respectively, they report mean rank and Hits@10 accuracy of 263 and 75.4% on WN18 as well as 243 and 34.9% on FB15K.

While TransE still remains one of the notable works in relation prediction, it is known to have problems in modelling reflexive, one-to-many, many-to-one and many-to-many relations. To model these relations more effectively, Wang et al. (2014) implemented *TransH* which projects both entities (of a 3 element triple, as above) into a relation specific hyperplane where they are connected together by a translation vector of that relation. The model enables the entities to have slightly different roles (via projection) according to the relations that hold them, and thus, learns the relations better. On an identical test setup as above, TransE performs better on WN18 than TransH (mean rank and Hits@10 accuracy of 318 and 75.4% respectively). However, on the FB15K dataset, TransH has the best performance (as compared to TransE and the other baselines) with a mean rank and Hits@10 accuracy of 211 and 42.5% respectively. Moreover, they report a substantial increase of about 12% on 1-to-many, many-to-one and many-to-many relations on the FB15K data, as compared to TransE.

Lin et al. (2015) too propose an improvement over TransE and one of its successor *TransR* by implementing *TransR*. Their model first embeds entities and relations in separate semantic spaces and then projects the entities into the relation space through mutually exclusive parameters for each relation. Their training objective remains similar to that of TransE regarding the nearness of the tail entity vector to a combination of the head entity vector and the relation vector. They test their model (against their implementation of TransE) on WN11 and FB15K and report an improvement of 11.2% and 6.8% the former and latter dataset respectively.

Since all of these works (and their subsequent extensions) use knowledge-base graphs for modelling purposes, they tend to design their models around the 3 element triple format. Our model too is a variant of this triple-based format.

However, these works consistently work with a fixed subset of knowledge-base datasets. Since the focus of these works is on increasingly optimized knowledge prediction, a fixed subset leads to standardized baselines as well as easier reproducibility of previous work, when required. On the

other hand, our focus is on understanding the factors that affect relation prediction and not so much on model optimization. For this reason, our models are domain specific, and consequently, we create our own datasets that help us explore the entire range of relations (and, a diverse set of related entities) within the same domain.

Lastly, these works also tend to use evaluation measures (like, Hits@x – defined in Section 6.3.3) that are biased towards one-to-many relations (or generally, multi-relational data) in the sense that relations with multiple values (correct answers) tend to get a higher score. We, on the other hand, use a simpler evaluation measure that allows us to assess one-to-one and multi-relational data in the same light.

6.3 Experimental Setup

We design this experiment as a supervised learning task with the aim to assess the extent to which distributional representations of categorical relations can be predicted from a distributional semantic space.

More concretely, we define two models that, given a distributional vector of a target entity '*t*' (*Star Wars*), a symbolic categorical relation 'ρ'[1] (*director*) and the corresponding relatum '*r*' (*George Lucas*), predicts the distributional vector for the relatum (plural: *relata*).

In the next two subsections, we first describe the models. Thereafter, we describe the evaluation metric and the baseline in the subsequent subsections.

6.3.1 Model 1: The Linear Model (LM)

The Linear Model (*LM*) is inspired by the "phrase analogy" evaluation of distributional word representations by Mikolov et al. (2013c): The authors show that semantic regularities and relationships are present as vector offsets and, all pairs of words that share a particular relation are

[1] The greek alphabet 'rho'.

also related by the same constant offset. Therefore, if $(\overrightarrow{man} - \overrightarrow{woman} = \overrightarrow{king} - \overrightarrow{queen})$ in a semantic space, then the vector $\overrightarrow{king}$ can be estimated from: $\overrightarrow{queen} + (\overrightarrow{man} - \overrightarrow{woman})$.

We extend the above analogy for categorical relation prediction as follows: consider the categorical relation `Continent(Germany,Europe)`. Our *LM* model, by its design, will predict the relatum `Europe` as shown in the equation:

$$\overrightarrow{Europe} = \overrightarrow{Germany} + ((\overrightarrow{Europe} - \overrightarrow{Italy})_1 + ... + (\overrightarrow{Asia} - \overrightarrow{India})_N)/N \tag{6.1}$$

Where, the N (relatum-target) pairs come from the training set. On the basis of the conclusions drawn by Mikolov et al. (2013c), we expect the LM model to make reasonable predictions on the assumption that categorical relations are represented additively in the distributional space. Thus, the relatum vector is the sum of the target vector and prototype of the categorical relation. Recall from Experiment 2, in Chapter 4, that we have earlier shown prototype vectors to be good representations of concepts as compared to their corpus-based representations in the distributional space.

Defining the LM model more formally: Given a set of triples that instantiate a relation ρ in the training set: $T_\rho = \{(t_i, \rho, r_i)\}$, the *LM* model learns the distributional representation of a relatum $\hat{r}$ of an input (t, ρ) by summing the target vector t and the averaged (or centroid) difference vector of the other (relatum – target) pairs in the training set:

$$\hat{r}(t, \rho) = t + \sum_{(r,\rho,t) \in T_\rho} (r - t)/N \tag{6.2}$$

6.3.2 Model 2: The Nonlinear Model (NM)

As depicted in Figure 6.1, the nonlinear model (*NM*) is a feed-forward neural network that introduces nonlinearity through a hidden layer, which transforms composition of target and relation, from which the relatum can

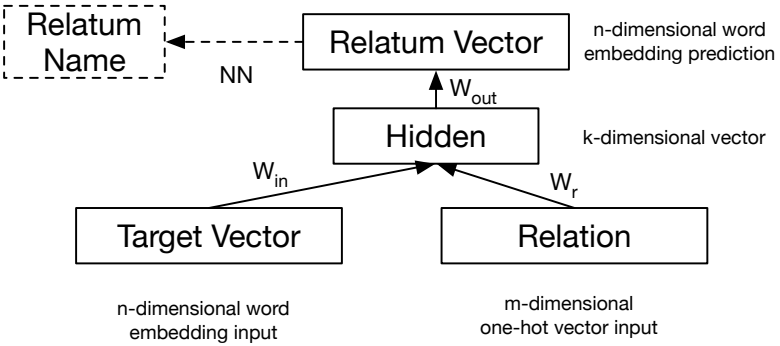


FIGURE 6.1: Nonlinear model (NM) structure

be predicted. As described in Equation 6.3, given an input of a target entity and a categorical relation (as vectors) the relatum vector is predicted as:

$$\hat{r}_\theta(t,\rho) = \sigma(\sigma(t \cdot W_{in} + v_\rho \cdot W_r) \cdot W_{out}) \tag{6.3}$$

Where, v_ρ is the relation encoded as an m-dimensional one-hot vector and the three matrices W_{in}, W_r, W_{out} form the model parameters θ. For the nonlinearity σ, we use $tanh$.

The *NM* model, unlike the *LM* model described above, can theoretically make accurate predictions even if relations are not additive in embedding space. Also, its sharing of training data among relations should lead to more reliable learning for infrequent categorical relations.

The model design is inspired by Levy and Goldberg (2014b) and similar to models used in knowledge base completion (for example, Socher et al., 2013a). However, our *NM* model requires only the target distributional vector and the relation vector as an input during decoding, which makes it much more efficient than the model by Socher et al. (2013a) (as described in Section 5.2).

$$L(\theta) = \sum_{(t,r)} (\cos(\hat{r}_\theta(t,\rho), r) - \alpha \cdot \cos(\hat{r}_\theta(t,\rho), nc(\hat{r}_\theta(t,\rho)))) \quad (6.4)$$

Typically, the objective functions of related work aim to minimize the distance between the predicted and the correct relatum (Socher et al., 2013a) as well as, penalize the model on negative samples (Bordes et al., 2013; Wang et al., 2014). While we take inspiration from these approaches, we further optimize the objective function as listed in Equation 6.4, where $nc(v)$ is the *nearest confounder* of v, i.e., the next neighbor of v that is not a relatum for the current target-relation pair. Thus, we *minimize* the cosine distance between the predicted vector $\hat{r}$ and the gold vector r for the relatum while *maximizing* the cosine distance of the prediction to the closest negative sample. We believe that the latter is a critical step in model learning because at each iteration it pushes the predicted relatum away from highly similar examples, thereby, ensuring that the model optimizes only on the correct answer without being thrown off by high semantic similarity. For instance, consider the positive example `Capital(Germany, Berlin)`: The loss function will minimize the distance between the predicted relatum and *Berlin* and, maximize its distance from nearest neighbours of Berlin, like *Munich, Frankfurt, Hamburg, Stuttgart, etc.* – the other large, multi-cultural cities of Germany which are quite similar to Berlin.

We also introduce a weight $\alpha \in [0,1]$ for the negative sampling term as a hyper-parameter optimized on the development set. During training, we apply gradient descent with the adaptive learning rate method AdaDelta (Zeiler, 2012).

6.3.2.1 Hyperparameter tuning

We adopt the best AdaDelta parameters from Zeiler (2012), viz. $\rho = 0.95$ and $\epsilon = 10^{-6}$. We optimize the negative sampling weight α (in equation

6.4) by line search with a step size of 0.1 on our largest domain[2]. We find 0.6 to be the optimal value for α, which we reuse for all domains. Due to the varying dimensionality m of the relation vector per domain, we set the size of the hidden layer to $k = 2n + m/10$ (n is the dimensionality of the word embeddings, see Figure 6.1). We train all models for a maximum of 1000 epochs with early stopping.

6.3.3 Evaluation

As described above in Section 6.3.1 and 6.3.2, our models *LM* and *NM* predict vectors (relata) in a continuous vector space. Consequently, one cannot expect these (or, any other) models to predict the output vector precisely. Therefore, to be able to evaluate model performance, we first have to label the predicted vectors. To achieve this, we apply *nearest neighbor mapping* using the set of all unique targets and relata to identify the correct relatum name. Simply put, the label of the predicted vector is the label of its nearest neighbour in the distributional space.

We then perform a ranking based evaluation. Note that, as mentioned in Section 6.2, the popular evaluation measure for such tasks is to report accuracy on *Hits@x*, where x (usually 3, 5 or 10) indicates the proportion of correct relata (or generally, entities) ranked in the top-x list. Due to the leniency of this evaluation measure towards one-to-many relational evaluation, we forgo this method and instead choose an Information Retrieval-style ranking evaluation.

We compute the rank of the correct relatum r, given the target t and the relation ρ, in the test set T and aggregate these ranks to compute the *mean reciprocal rank* (MRR):

$$MRR = \frac{1}{||T||} \sum_{(t,\rho,r)\in T} \frac{1}{rank_{t,\rho}(r)} \tag{6.5}$$

[2]We will discuss the dataset design in more detail in the following section.

where *rank* is the nearest neighbor rank of the relatum vector r given the prediction of the model for the input t, ρ. We report results at the relation level as well as macro- and micro-averaged MRR for the complete dataset.

6.3.4 Baseline (BL)

We compute a frequency based baseline. For each test datapoint triple $i = (target_i, relation\ \rho_i, relatum_i)$, we compute the baseline rank MRR_{ρ_i} by computing the multiplicative inverse of the rank of $relatum_i$ from the training data. The rank of $relatum_i$ is its position in the frequency-ordered list of all the relata of the relation ρ in the training set. Next, we compute MRR_{ρ} by averaging all instances of MRR_{ρ_i} in the test set.

Recall that the motivation of this experiment is to build a model that can predict on unseen data and, the baseline above mirrors this motivation. In other words, the baseline expects that not every pair of (relation ρ_i – $relatum_i$) in the test set is observed in the training data; in which case, the MRR_{ρ_i} for a given test datapoint is 0. Since our model evaluation is dependent on the predicted relatum, our baseline model too does not consider the target.

6.4 Datasets

In this experiment, we follow the same data extraction and dataset design methodology that we adopted in the previous chapter.

We extract all our data from Freebase, which maintains attribute-values of entities as tuples, automatically without any filtering via manual inspections. However, unlike the last experiment where we extracted data for only two domains *countries* and *cities*, this time we extract data from 7 most populous (largest) domains of Freebase which are also quite diverse in the types of entities that they maintain: *animal, book, citytown, country, employer, organization, people*.

Domain	No. of Relations (ρ)	No. of Target–Relatum pairs
animal	24	3,428
book	22	7,014
citytown	46	86,551
country	89	191,196
employer	76	14,658
organization	53	8,989
people	91	11,397
Total	402	323,233

TABLE 6.1: Extraction statistics for 7 Freebase domains by relations ρ and (target–relatum) pairs per relation

For each domain, we extract categorical relational information in the form of triples within two hops in the Freebase graph: (*entity, attribute, value/entity*). We perform no additional filtering other than to remove all numeric relations. To avoid bias in model learning due to a skewed distribution of relations, we employ a relation thresholding. We limit very large relation types to a maximum of 3000 with random sampling and, we remove those relation which have fewer than 3 datapoints (or, relatums).

Table 6.1 shows the results of the data extraction process. For each domain, we list the total number of unique categorical relations ρ that we have extracted (in the first column) along with the total number of unique (target–relatum) pairs in the second column. The overall distribution of relations is quite diverse with a maximum of 91 for *people* and minimum of 22 for *book* domains. Similarly, the extraction resulted in a maximum of about 200K (target-relatum) pairs for the *country* domain and a minimum of 3,428 pairs for the *animal* domain. This results in a quite challenging dataset that demonstrates the generalizability of our models and is comparable, in variety and size, to the *FB15K-237* dataset (see Table 5.2 in Section 5.2).

We create 7 datasets, one for each domain, where the distributional

representations for the targets and relata come from our Freebase-based distributional space, as described in Section 3.3.1 in Chapter 3. Each representation is 1000-dimensional which has been constructed from the 100B token Google News corpus and is L_2 normalized i.e. a unit vector. We retain only those datapoints where both target and relatum are covered in the distributional space and, the numbers reported in Table 6.1 are based on the datapoints where we were able to acquire a mapping.

Finally, we split all datasets into training, validation, and test sets (60, 20, 20%) respectively. The split applies to each relation type: in test, we face no unseen relation types, but there are unseen datapoints for each relation[3]. That is to say that if `(Germany, Capital, Berlin)` were to occur as a test datapoint, then the relation `Capital` occurs in the training data, but not with `Germany - Berlin` as its target and relatum pair.

Note that from our 7 domains, *citytown* and *country* are analogous, but *not* identical, to the *city* and *country* domains in Chapter 5. Recall that according to Section 5.4.1, the two domains had both numeric and categorical relations with a relation thresholding of 15 – which resulted in a comparatively smaller subset of categorical relations[4]. In this chapter, we follow a different thresholding mechanism due to the experiment design as well as keep only those categorical attributes which have a mapping to our semantic space – thus, leading to a much larger set of categorical relations. Moreover, *citytown*, in addition to cities, also includes information about small towns and administrative districts which are not large enough to be categorized as cities. Overall, in terms of data, both the domains in the two chapters are not comparable – despite having similar names, .

[3]The dataset are available at: http://www.ims.uni-stuttgart.de/data/ RelationPrediction.html

[4]247 and 106 categorical relations (or, binary attributes) in the *country* and *city* domain respectively.

6.5 Results

Table 6.2 shows the MRR scores of our models on the 7 datasets; one for each domain. We first list the Baseline (BL) scores, followed by the scores of the Linear model (LM) and then the Nonlinear model (NM). We also list the averaged micro and macro MRRs for each of the model at the bottom.

The overall results clearly show that predicting categorical relations (i.e., predicting distributional representations of categorical values) is a non-linear problem. The nonlinear model *NM* consistently gives the best results and statistically outperforms the linear model *LM* on all domains according to a Wilcoxon test (α=0.05) (Wilcoxon, 1992). In turn, both *LM* and *NM* clearly outclass the baseline BL.

The MRR based evaluation has a range between $[0..1]$ and looking at the model scores in the table, one might initially perceive these scores to be low. However, recall that MRR is the multiplicative inverse of the rank of the first correct answer i.e., between $[0..1]$ a score of 1 indicates that the prediction has rank 1, 0.5 indicates rank 2, 0.33 indicates a rank 3 and so on. Since our task requires the models to predict a 1000 dimensional real-valued distributional representation, consequently, it is non-trivial for our models to estimate the correct relatum in an open-vocabulary space of tens of thousands of words. In this light, the (macro) averaged MRR scores of *NM*- 0.25 and *LM*- 0.16 tell us that the distributional representations predicted by our models are around the fourth and the sixth nearest neighbours of the correct relatum. While we find this outcome to be reasonably satisfactory, it is nonetheless surprising because the model has been tuned on the *country* domain dataset on which, incidentally, we report the worst results.

Our only outlier, in terms of performance, is the *country* dataset (*NM*- 0.18 and *LM*- 0.08) which not only has a high number of relations (ρ = 89) but also more than double the number of *target-relatum* datapoints (191,196) than our second most populous domain (*citytown* – 86,551). The high number of datapoints implies that atleast some relations (if not all) have a much higher number of relata per target. This in turn implies that

Domain	BL	LM	*NM*
animal	0.11	0.16	**0.29**
book	0.11	0.24	**0.26**
citytown	0.05	0.13	**0.26**
country	0.04	0.08	**0.18**
employer	0.05	0.15	**0.23**
organization	0.07	0.17	**0.26**
people	0.09	0.19	**0.27**
Micro average	0.06	0.14	**0.22**
Macro average	0.08	0.16	**0.25**

TABLE 6.2: Baseline (BL) performance as well as Linear (LM) and Nonlinear (NM) MRRs on 7 datasets (each representing a domain), with micro-marco averages.

the models, in their current design, plausibly have a hard time learning multi-valued categorical relations.

Note that the difficulty of predicting multi-valued categorical relations, specially when the prediction is a real-valued vector, is not un-recognized in the knowledge representation community. This is one of the reasons for the popularity of evaluation measures, like `Hits@x`, which awards a positive score when the correct result is within a 'list' of `x` elements. However, not only does this evaluation pushes the overall results in a higher range but, it also diverts the focus towards problems of frequency. That is to say that since relations with higher relata per target are found to score better than those with fewer relata per target, the focus is turned on analyzing model performance of relations of the latter type. As a consequence, for instance, data sparsity gets prioritized over other potential culprits behind poor model performance.

Our initial results are indicative of relations with few relata per target doing well: The above average MRR scores in comparatively sparser domains, 0.29 and 0.26 for *animal* and *book* respectively show that our models are (comparatively) able to generalize better when relata per target

are rather limited. Moreover, the closeness of our micro and macro averaged MRR results provides an additional support to the generalizing capability of our models. On the other hand, relations with many relata per target perform poorly vis-a-vis the MRR scores of the country domain. This is contrary to the conclusions of the related works (Section 6.2) which report a better performance of one-to-many relations over one-to-one relations.

6.6 Analysis

In this section, we present a detailed empirical analysis to reflect on the conclusions drawn above and, further investigate the effect of diverse frequency distributions on model learning as well as the factors that are responsible for good (or, bad) relation prediction.

Analysis at relation level: Table 6.3 shows the good and bad relations, with MRRs greater than 0.3 and less than 0.1 respectively. While the numbers vary across domains, the models tend to do badly on around 40-50% of all relations, and obtain good scores for less than one third of all relations.

Figure 6.2 shows the distribution for the best domain (*animal*) and the worst one (*country*). Both plots show a Zipfian distribution (see footnote on Page 21 for details) with a relatively small set of well-modelled relations and a long tail of poorly modelled ones. *NM* does better or as well as *LM* for almost all relations. The performances of the two models are very tighly correlated for difficult relations; they only differ for the easier ones, where the nonlinear model *NM* evidently captures the data better.

Qualitatively, the two models differ substantially with regard to prediction patterns at the level of targets. Table 6.4 shows the first predictions for three targets from two relations: *continent*, where *NM* outperforms *LM*, and *capital*, where it is the other way around. The errors of *NM* consist almost exclusively in predicting semantically similar entities of the correct relatum type, e.g., predicting Quito (the capital of Ecuador) as capital of Venezuela.

Domain	Good Relations $MRR_\rho > 0.3$	Bad Relations $MRR_\rho < 0.1$
animal	38%	42%
book	9%	68%
citytown	28%	39%
country	20%	52%
employer	30%	45%
organization	34%	42%
people	34%	23%

TABLE 6.3: Test set statistics of categorical relations on the NonLinear (NM) model: Percentage of relations with good and bad MRRs.

Relation	Target	Correct	*LM*	*NM*
continent	Japan	Asia	Japan	**Asia**
	Kazakhstan	Asia	Central Asia	**Asia**
	Nicaragua	North America	Latin America	Americas
capital	Nepal	Kathmandu	Nepal	Dhaka
	Qatar	Doha	Qatar	Riyadh
	Venezuela	Caracas	**Caracas**	Quito

TABLE 6.4: Example predictions for two `country` relations (correct answer in boldface)

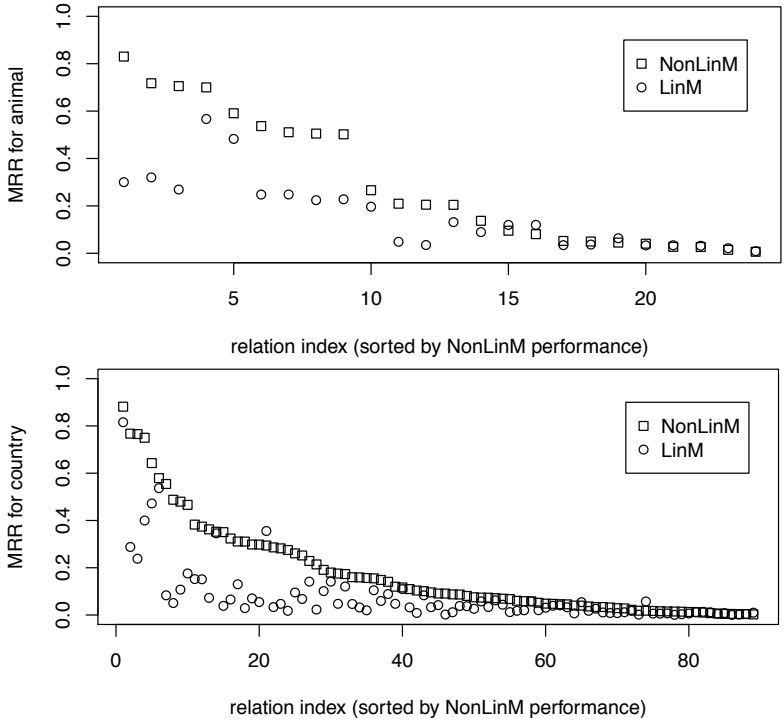


FIGURE 6.2: Results by relation for best and worst domains (*animal*, above; *country*, below), sorted by *NM* performance

In contrast, the *LM* model has a harder time capturing the correct type, predicting country entities as capitals (e.g., Nepal as the capital of Nepal).

Note that since we extract all Freebase data automatically without manual filtering (Section 6.4), we notice that some relation triples are questionable (albeit, not incorrect). For example, *Nicaragua* is listed on North America. This is not inaccurate *per se* because on a global scale the land-masses are indeed divided between North and South America and, Nicaragua lies in the former. However, in discourse and daily usage Nicaragua is more commonly assumed to be a part of the Central America and moreover, countries including Mexico and those south of Mexico have

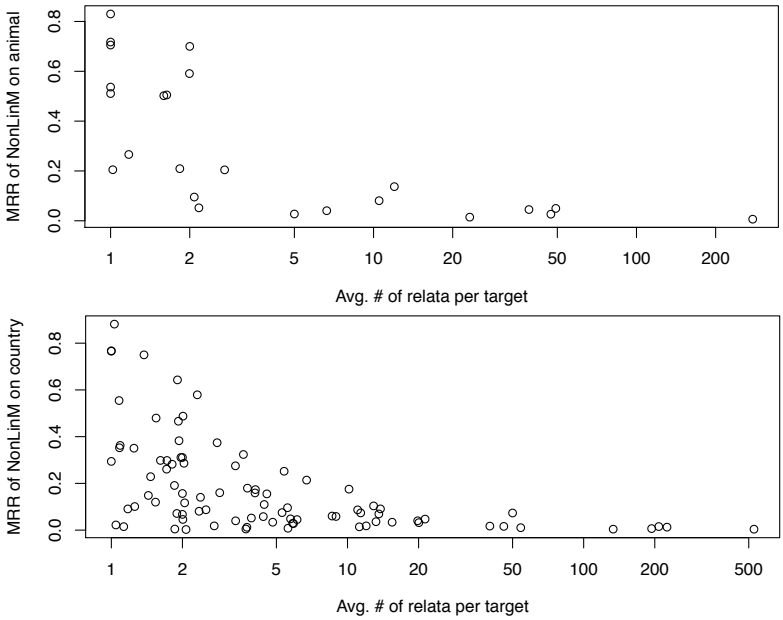


FIGURE 6.3: Scatterplot: MRR of *NM* vs. number of relata per target (above: *animal*, below: *country*)

socio-culturally more in common with their South American counterparts as compared to USA (for a detailed discussion on conceptual biases see Chapter 5).

Analysis of Difficulty: So what makes many Freebase relations hard to model? To test for sparsity problems, we first computed the correlation between model performance and the "usual suspect" relation frequency (number of instances for each relation). In NLP applications, this typically yields a high positive correlation. The first column (Targets per Relation) of Table 6.5 shows that this is not true for our dataset. We find a substantial positive correlation only for *people*, correlations around zero for most domains, and substantial negative ones for *organization* and *country*. For

Domain	Targets per Relation	Relata per Target
animal	.07	-.34
book	.09	-.15
citytown	-.12	-.22
country	-.32	-.23
employer	.01	-.35
organization	-.24	-.29
people	.23	-.25

TABLE 6.5: Test set statistics of categorical relations on the NonLinear (*NM*) model: Spearman correlation of MRRs with frequency of Targets per Relation and Relata per Target.

these domains, therefore, frequent relations are actually harder to model. We further identify two main sources of difficulty:

(1) One-to-many relations. As hinted in Section 6.5, relations with many datapoints tend to be *one-to-many*. We assume this to be a major source of difficulty, since the model is presented with multiple relata for the same target during training and will typically learn to predict a centroid of these relata. As an extreme case, consider a relation like `administrative_divisions` that relates the US to all of its federal states: the resulting prediction will arguably be dissimilar to every individual state.

To test this hypothesis, we computed the rank correlation at the relation level between the number of relata per target and *NM* performance, shown in the second column (Relata per Target) of Table 6.5. Indeed, we find a strong negative correlation for every single domain. This is also graphically depicted in Figure 6.3, that plots relation performance (y axis) against the ratio of relata per target (x axis: one-to-one on the left, one-to-many on the right) for *animal* and *country*. Both plots show poor MRR scores for one-to-many relations.

Relation	BL	LinM	NonLinM
tournaments	0.88	0.82	0.88
continent	0.29	0.29	0.77
country	0.25	0.24	0.77
⋮			
disputed_territories	0.00	0.01	0.01
horses_from_here	0.00	0.01	0.01
2nd_level_divisions	0.00	0.00	0.01

TABLE 6.6: The three most easy and most difficult relations for the country domain

Qualitatively, Table 6.6 shows examples for the three most easy and difficult relations for the *country* domain. The list suggests that relations tend to be easy when they associate targets with single relata: the relation country maps territories and colonies onto their motherlands, and the tournaments relation is only populated with a few Commonwealth games. In contrast, relations that map targets on many relata are difficult, such as administrative divisions of countries, or a list of disputed_territories. Note that this is not an evaluation issue, since MRR can deal with multiple correct answers. Our models do badly because they lack strategies to address these cases.

(2) Lack of contextual support: One-to-many relations are not the only culprit. Strikingly, Figure 6.3 shows that a low target-relatum ratio is a *necessary* condition for good performance (the upper right corners are empty), but not a *sufficient* one (the lower left corners are not empty either). Some relations are not modelled well even though they are (almost) one-to-one. Examples include currency_formerly_used or named_after for *country* and place of origin for *animal*. Further analysis indicated that these relations suffer from what we called *lack of contextual support* in Section 5.6.1: Although they are expressed overtly in the linguistic context of the target and relatum (and often even frequently so), their realizations cannot be tied to individual words or patterns because these patterns are relatively specific, such as predicate-argument structures like *X used to*

pay with Y, X is named in the honor of Y. Such structures are hard to pick up by word embedding models that make the bag-of-words independence assumption among context words.

6.7 Adaptation of related models to our experiment

As mentioned in Section 6.2, our models are variants of other state-of-the-art models in the sense that these too employ the use of 3 element triple format for attribute modelling (be it an *entity* or a *relation*). While we do not compare these models to our work for aforementioned reasons, we however hypothesise in this section as to how these can be gainfully employed for categorical attribute prediction within the bounds of our current semantic space and datasets.

We start with *TransE* (Bordes et al., 2013), that assumes that the following relation holds: $head + relation \approx tail$ (where, {head, tail} ϵ entity). This model is analogous to our linear model (in Section 6.3.1) in the sense that this model estimates the tail (categorical attribute:*Europe*) from a function of its head (*Germany*) added to the aggregated relation vector (categorical relation:*capital* – which we compute as a centroid). We, just as Bordes et al. (2013), assume that semantic relations are represented as a translation in the embedding space. Our non- linear model (in Section 6.3.2) is conceptually similar to *TransE* where we concatenate the *head* and *relation* embeddings as input to estimate the *tail* embedding.

The *Single Layer Model* proposed by Socher et al. (2013a) gives the concatenation of the *head* and *tail* embeddings as input to a non-linear model to predict a relation embedding. This model can be easily adapted to our datasets where we label the predicted relation embedding according to its nearest relation embedding (one-hot vector) in our setup.

The novelty of *TransH* by Wang et al. (2014) lies in the concept that *head* and *tail* embeddings are different for multiple relations and, can be

obtained by projecting them into different relations specific hyperplanes. *TransH* can be adapted for our task by first projecting *head* and *tail* embeddings as concatenation of *head* + *relation* and *tail* + *relation* embeddings, respectively, by a 1-hidden-layered non- linear model. Thereby, we can extract the hidden layer which will represent relation specific *head* and *tail* embeddings. In the same spirit, *TransR* can be adapted for our task by first projecting the *head* and *tail* entity embeddings (from our distributed space) and relation embeddings into distinct spaces and then projecting the entities into their relation specific semantic space. The resulting *head* entity and relation embeddings can then be supplemented as inputs to our non- linear model (in section 6.3.2) which in turn can predict the categorical attribute i.e., the *tail* entity embedding.

Suffice to say (from the above) that most models built around the 3 element triple format can be adapted to our task with: 1) slight changes to their architecture; or, 2) transformation of our semantic space according to their methodology and subsequently using the transformed space into our task setup. The evaluation methodology (in section 6.3.3) can remain the same to assess different model performances.

6.8 Conclusion

As an extension of Experiment I (Chapter 5), in this chapter, we alleviated the short-comings of predicting categorical relations by implementing a system that predicts distributional representations of categorical values. To predict the representations, we described two models – one linear and the other non-linear. The task is challenging because we predict 1000 dimensional real-valued distributional word representations and, contrary to many NLP tasks our results show that high difficulty in estimations is not due to low frequency.

Our results yield insights into how the distributional space that we work with represents entity relations: they are generally not represented additively, and nonlinearity helps. Our analyses complement the insights on

the behavior of numeric attributes of entities in Chapter 5: that categorical relations, like numeric attributes, are difficult to model if they are not specifically expressed in the lingusitic context of target and relatum. A new challenge specific to categorical relations are situations where a single target maps onto many relata.

If none of the two problems above applies then relations are *easy* to model. If either one of them applies, they are *difficult*. And if both apply, they are essentially *impossible*.

Among the two problems, the problem of one-to-many relations appears easier to address, since a continuous output vector can be similar to many relata; at least in principle. One alternate is to use syntax-based distributional representations (Levy and Goldberg, 2014a) that can better pick up the specific context patterns characteristic for these relations. Another approach is to optimize the input by jointly training representations constructed from corpus evidence as well as knowledge base graph (Perozzi et al., 2014; Toutanova et al., 2015).

We pursue the latter approach and propose a simple feed-forward neural architecture to jointly predict numeric and categorical attributes (Thejas et al., 2019). Due to the correlations among attributes of different kinds, joint prediction should be an improvement over individual prediction. Our experiments on seven Freebase domains (starting from the datasets described in this chapter) find partial support for our hypothesis: the joint model leads to a substantial improvement for numeric attribute prediction, however, the model performance remains largely unchanged for categorical attributes.

Chapter 7

Conclusion and Future work

This thesis is devoted to investigating entities in a distributional setup.

One of our main contributions was to empirically show that in distributional semantics, meaning representations of entities (instances denoted by proper nouns) are different from those of categories (concepts denoted by common nouns). Our experiments and analyses point towards the differences being semantic (i.e., based on type). However, at this point, we also do not explicitly rule out the possibility that the differences could also arise due to their morpho-syntactic properties (based on parts of speech).

Our support towards the differences being semantic, as observed while contrasting the distributional behaviour and geometry of entities and categories in Chapter 3 and 4, comes from the fact that we could successfully establish the presence of fine-grained information within the distributional representations of entities in Chapter 5. For instance, the *GDP* and *geolocation* of *Germany* is *3.6 trillion* and *Latitude: 51.1642; Longitude: 10.4541* respectively. This is our second key contribution.

The existence of fine-grained information, that we systematically extracted from entities, allowed us to formally (and empirically) establish that the encoding of fine-grained information is the reason behind entities being successfully modelled in tasks related to knowledge prediction (or, completion) – despite the existence of widely accepted antithetical assumptions in distributional semantics (as discussed in Chapter 1). Through our analyses, we also discovered that not all types of fine-grained information is effectively captured in distributional representations of entities. In Chapter 5 and 6 we identified factors, other than data sparsity, that we believe are

responsible for fine-grained information being captured and extracted from distributional representations.

One of these factors that requires a specific mention is the *contextual support* (or lack of it) for the fine-grained information in text. As per our analyses, we found contextual support to be as critical a factor for good model predictions as frequency i.e., those fine-grained relations which we found to be contextually well supported are easier for the models to learn. On the other hand, relation frequency did not apparently have that pronounced an effect on our results as reported by other studies (Bordes et al., 2013; Wang et al., 2014).

An interesting find from our study of fine-grained information was the encoding of socio-economic and cultural regularities by distributional models (Chapter 5). While the existence and use of such regularities are well established in cognitive literature, we found that our distributional semantic space had captured these regularities as well. We saw that our models predicted attributes like `geolocation`, `GDP`, `` $CO_2$ emissions ``, etc. to a high degree of accuracy despite the fact that one would not typically expect these attributes to be captured by the distributional models. We believe that such attributes could be successfully learned because the models could effectively extrapolate from these encoded regularities, which at a superficial glance might be seemingly unrelated. We interpret these observations as further support towards the claim that distributional models mirror human learning to some extent.

Finally, through our experiments, we also found that when it comes to modelling entities and categories together, via the lexical relation of instantiation, entity-based representations of categories are more suitable as compared to the representations created from observations of the category denoting noun (Chapter 4). This also confirmed that entity representations, when averaged, can be understood to serve as 'prototypes' of the categories they represent.

A possible future work, that has its motivation in the fact above, is the extension of our approach to account for not just for words or phrases

directly denoting an entity (*India*), but also other kinds of singular and plural definite descriptions (*the largest Hindi speaking nation in the world* or *the cities in Western Germany*). It is entirely plausible that certain categories (or, concepts) are also as specific as entities and, therefore, can be used to model fine-grained knowledge prediction as well – an aspect that we have not explored in the thesis but we believe is worth investigating.

Another extremely interesting (and challenging) extension is to tackle entities in private discourse and those that are anonymous (for example, *the bird we saw this morning*) for which standard distributional techniques are seemingly inefficient. It is still not clear where we could gather appropriate training data to learn about the specific properties of such entities. A plausible solution is to induce coreference information about such entities into their distributional meaning representations by using mined co-reference chains in addition to co-occurrence information. Initial work in this direction has shown that such representations are better for semantic tasks as compared to those constructed purely out of co-occurrence information (Adel and Schütze, 2014). However, more rigorous analyses are required to assess if the effectiveness of such an approach is generalisable (or not) and, what are its limitations. Note that such coreference based representations could not only lead to more semantically precise representations of entities but could also help in mitigating the impact of morpho-syntactic information that gets encoded in the distributional representations (as pointed out above).

We believe that the foundations that we have laid can further profit from more work towards better assessing the properties of instances as well as the effects of design factors such as: more sophisticated models, the choice of the underlying semantic space and dataset construction. Our choice of semantic space was determined by the fact, that at the time of this work, this space provided us with a reasonable coverage of entities (and their corresponding categories).

Recent times have brought forward novel meaning construction models that can construct contextual embeddings – like BERT (Devlin et al., 2019),

and subword embeddings – like FastText (Bojanowski et al., 2017) which have been show to be state-of-the-art at many computational linguistics tasks and can be compared to the static embedding models – like Word2Vec. Note that, over time, it has been empirically shown that BERT embeddings generally outperform subword and static embeddings in most CL and NLP tasks (Ethayarajh, 2019). One reason for restricting the scope of our current explorations in this direction is that at the time our experiments were done, to the best of our knowledge, our semantic space consisted the largest collection of entities that were also mapped to a knowledge base (Freebase). Additionally most of the models at the conclusion of this work would have difficulty dealing with such a large collection of proper nouns (as ours). Also, chronologically, most of the newer models (subword and contextual) were introduced after the research work was concluded. But more importantly, since our data consists mainly of lexical relations held between lemmas (or types), using subword models that excel at capturing morphological information will plausibly not result in an improvement; at the same time, using BERT – which is a token-based sequence model and better tuned towards common nouns (and not proper nouns) – might be better but not significantly.

To incorporate such models into our scope of work (for contrast) we would be either required to collect corpora that covers the entities in our work and then use these meaning construction models to either construct or fine-tune these embeddings. For instance, in FastText, Bojanowski et al. (2017) construct word embeddings using their character n-gram representations. So, *Washington* might be constructed from 3-level-n-gram representations: was, hi, ing, ton. In case of BERT, the problem is all the more difficult because we would be required to design and create sequence-based datasets (comparable to all our datasets) that can be ingested by the models. That is to say that we would have to convert all lemma based datapoints to sentence/phrase based structures (or, entity descriptions). Therefore, while we leave this endeavour to future work, we present a brief update on how such an objective can be achieved by discussing related

work done in this direction.

Recall that the question we want to focus on is – whether contextual embeddings will be necessarily better than static embeddings? In FastText, Bojanowski et al. (2017) state that when it comes to *seen* words, FastText embeddings are just marginally better (and, sometimes equivalent or worse for some datasets) than static embeddings. As evident from the example above (of *Washington*), character based n-gram construction is seemingly not an effective strategy. When it comes to BERT, the KBC research community has attempted to use it as a main-stream model for knowledge prediction. Yao et al. (2019) fine-tune the BERT model by giving a triple (entity_1, relation, entity_2) as a sequential input to predict the plausibility of a triple. They also construct models to predict a relation given two entities or an entity given another entity and a relation[1]. The triples (in the former) and the entities (in the latter) are token sequences, like *Steve Jobs*, as well as descriptions of the entities, like *Steve Jobs was an entrepreneur* and, the model input is the summed representation of the token embeddings, the segment and the positional embeddings. They compare their work with other state-of-the-art models mentioned in Section 6.2 by using the datasets mentioned in Section 5.2. When it comes to triple plausibility classification, they attain a increment of 4-10% over models using static embeddings. This makes sense because triples can be viewed as *sequences* specifically in a multi-word token setup. For the same reason, the relation prediction task also reports a similar improvement. However, when it comes to entity prediction their models are only marginally better and below par when evaluated on Wordnet and Freebase datasets respectively, using *Hits@10*. Kim et al. (2020) show that this is because the BERT model fails to efficiently capture relational information and additionally gets confounded with semantic similarity in case of multi-word representations of entities and relations. They improve upon the results of Yao et al. (2019) by using an ensemble model. Echoing a similar approach but specifically on medical data, Nadkarni et al. (2021) show that ensemble models (which

[1]Recall that this is analogous to our tasks in chapter 5 and chapter 6

include BERT) significantly outperform others in entity/relation prediction tasks. However, Lenci et al. (2022) go on to show that when it comes to experimental data without context (such as the data in this work) then static models are empirically superior to contextual models. They also make a case against Ethayarajh (2019) and Bommasani et al. (2020), who show that contextual embeddings are better than their static counterparts, by arguing that BERT evaluation scores were better only because suboptimal static embedding models were used. Overall, as stand-alone models, there is ample evidence conclusively supporting our hypothesis that both subword and contextual models do not necessarily perform better than static models, specially with the type-based (and context deficient) data that we have used in this research.

In essence, this thesis also takes a small step towards bridging the gap that has existed between formal and distributional semantics. While formal semantics posits that entities and categories are distinct, distributional semantics on the other hand assumes (and treats) them to be similar. The latter is true to a certain degree due to their potentially shared contexts because of the cooccurrence based meaning construction methodology of distributional semantic models. However, our investigations revealed that the distributional distinctions between them are much more prominent than (earlier) expected due to the fine-grained information that is encoded in entity meaning representations. The take-away is that distributional modelling of entities requires a more focussed approach, both in terms of learning entity representations (so that their properties can be captured more effectively) and, applying these representations to downstream NLP applications that require the knowledge of these properties.

Bibliography

Abhishek, Ashish Anand and Amit Awekar. “Fine-Grained Entity Type Classification by Jointly Learning Representations and Label Embeddings”. In: *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain, 2017, pp. 797–807.

Adel, Heike and Hinrich Schütze. “Using mined coreference chains as a resource for a semantic task”. In: *Proceedings of EMNLP*. Doha, Qatar, 2014, pp. 1447–1452.

Aggarwal, Charu C and ChengXiang Zhai. “A survey of text clustering algorithms”. In: *Mining text data*. Springer, 2012, pp. 77–128.

Agichtein, Eugene and Luis Gravano. “Snowball: Extracting relations from large plain-text collections”. In: *Proceedings of the fifth ACM conference on Digital libraries*. 2000, pp. 85–94.

Aitken, Chris, Christine Stephenson, and Ryan Brinkworth. “Process classification frameworks”. In: *Handbook on Business Process Management 2*. Springer, 2010, pp. 73–92.

Alfonseca, Enrique and Suresh Manandhar. “An unsupervised method for general named entity recognition and automated concept discovery”. In: *Proceedings of the 1st international conference on general WordNet, Mysore, India*. 2002, pp. 34–43.

Alfonseca, Enrique and Suresh Manandhar. “Distinguishing concepts and instances in WordNet”. In: *Proceedings of the First International Conference of Global WordNet Association*. Mysore, India, 2002. URL: http://www-users.cs.york.ac.uk/{~}suresh/papers/DCAIIW.pdf.

Ali, Tariq, Sohail Asghar, and Naseer Ahmed Sajid. “Critical analysis of DBSCAN variations”. In: *2010 International Conference on Information and Emerging Technologies*. IEEE. 2010, pp. 1–6.

Aluç, Güneş, Olaf Hartig, M Tamer Özsu, and Khuzaima Daudjee. “Diversified stress testing of RDF data management systems”. In: *International Semantic Web Conference*. Springer. 2014, pp. 197–212.

Angeli, Gabor, Neha Nayak, and Christopher D Manning. “Combining natural logic and shallow reasoning for question answering”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 442–452.

Angeli, Gabor, Melvin Jose Johnson Premkumar, and Christopher D Manning. “Leveraging linguistic structure for open domain information extraction”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2015, pp. 344–354.

Armstrong, Sharon Lee, Lila R Gleitman, and Henry Gleitman. “What some concepts might not be”. In: *Cognition* 13.3 (1983), pp. 263–308.

Arora, Kushal, Aishik Chakraborty, and Jackie CK Cheung. “Learning Lexical Subspaces in a Distributional Vector Space”. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 311–329.

Arthur, David and Sergei Vassilvitskii. *k-means++: The advantages of careful seeding*. Tech. rep. Stanford, 2006.

Asuero, Agustin Garcia, Ana Sayago, and AG Gonzalez. “The correlation coefficient: An overview”. In: *Critical reviews in analytical chemistry* 36.1 (2006), pp. 41–59.

Auer, Sören, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. “Dbpedia: A nucleus for a web of open data”. In: *The semantic web*. Springer, 2007, pp. 722–735.

Baevski, Alexei, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. “Cloze-driven Pretraining of Self-attention Networks”.

In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 5363–5372.

Baldi, Pierre and Peter Sandowski. "Understanding Dropout". In: *Proceedings of Advances in Neural Information Processing Systems*. 2013, pp. 2814–2822.

Baroni, Marco, Eduard Barbu, Brian Murphy, and Massimo Poesio. "Strudel: A distributional semantic model based on properties and types". In: *Cognitive Science* 34.2 (2010), pp. 222–254.

Baroni, Marco, Raffaela Bernardi, and Roberto Zamparelli. "Frege in space: A program of compositional distributional semantics". In: *LiLT (Linguistic Issues in Language Technology)* 9 (2014).

Baroni, Marco, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. "Entailment above the word level in distributional semantics". In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2012, pp. 23–32.

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski. "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors". In: *Proceedings of ACL*. Baltimore, MD, 2014, pp. 238–247.

Baroni, Marco and Alessandro Lenci. "Distributional memory: A general framework for corpus-based semantics". In: *Computational Linguistics* 36.4 (2010), pp. 673–721.

Baroni, Marco and Alessandro Lenci. "How we BLESSed distributional semantic evaluation". In: *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. Edinburgh, UK, 2011, pp. 1–10.

Basile, Valerio, Soufian Jebbara, Elena Cabrio, and Philipp Cimiano. "Populating a knowledge base with object-location relations using distributional semantics". In: *European Knowledge Acquisition Workshop*. 2016, pp. 34–50.

Baxter, MJ. "Spatial k-means clustering in archaeology–variations on a theme". In: *Academia (Accessed February 17, 2017). https://www. academia. edu/18142974/Spatial_k-means_clustering_in_archaeology_–_variations_on_a_theme* (2015).

Bechhofer, Sean, Frank Van Harmelen, Jim Hendler, Ian Horrocks, Deborah L McGuinness, Peter F Patel-Schneider, Lynn Andrea Stein, et al. "OWL web ontology language reference". In: *W3C recommendation* 10.02 (2004).

Beltagy, Islam, Stephen Roller, Pengxiang Cheng, Katrin Erk, and Raymond J Mooney. "Representing meaning with a combination of logical and distributional models". In: *Computational Linguistics* 42.4 (2016), pp. 763–808.

Ben-David, Shai and Margareta Ackerman. "Measures of clustering quality: A working set of axioms for clustering". In: *Advances in neural information processing systems*. 2009, pp. 121–128.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. "A neural probabilistic language model". In: *Journal of machine learning research* 3.Feb (2003), pp. 1137–1155.

Bengio, Yoshua, Yann LeCun, et al. "Scaling learning algorithms towards AI". In: *Large-scale kernel machines* 34.5 (2007), pp. 1–41.

Berant, Jonathan, Andrew Chou, Roy Frostig, and Percy Liang. "Semantic Parsing on Freebase from Question-Answer Pairs". In: *Proceedings of EMNLP*. Seattle, WA, 2013, pp. 1533–1544.

Bergstra, James, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. "Theano: a CPU and GPU math expression compiler". In: *Proceedings of the Python for scientific computing conference (SciPy)*. Vol. 4. 3. Austin, TX. 2010, pp. 1–7.

Bhatt, Ganesh D. "Knowledge management in organizations: examining the interaction between technologies, techniques, and people". In: *Journal of knowledge management* (2001).

Bizer, Christian, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. "DBpedia – A crystallization point for the Web of Data". In: *Journal of Web Semantics* 7.3 (2009), pp. 154–165.

Blanco, Roi, Giuseppe Ottaviano, and Edgar Meij. "Fast and space-efficient entity linking for queries". In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM. 2015, pp. 179–188.

Blitzer, John, Fernando Pereira, Kilian Q Weinberger, and Lawrence K Saul. "Hierarchical distributed representations for statistical language modeling". In: *Advances in Neural Information Processing Systems*. 2005, pp. 185–192.

Bodenreider, Olivier. "The unified medical language system (UMLS): integrating biomedical terminology". In: *Nucleic acids research* 32.suppl_1 (2004), pp. D267–D270.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching word vectors with subword information". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146.

Boleda, Gemma and Katrin Erk. "Distributional semantic features as semantic primitives—or not". In: *2015 AAAI Spring Symposium Series*. 2015.

Boleda, Gemma and Aurélie Herbelot. "Formal distributional semantics: Introduction to the special issue". In: *Computational Linguistics* 42.4 (2016), pp. 619–635.

Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. “Freebase: a collaboratively created graph database for structuring human knowledge”. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 2008, pp. 1247–1250.

Bommasani, Rishi, Kelly Davis, and Claire Cardie. “Interpreting pretrained contextualized representations via reductions to static embeddings”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 4758–4781.

Bordes, Antoine, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. “Translating embeddings for modeling multi-relational data”. In: *Proceedings of Advances in neural information processing systems*. 2013, pp. 2787–2795.

Bordes, Antoine, Jason Weston, Ronan Collobert, and Yoshua Bengio. “Learning structured embeddings of knowledge bases”. In: *Twenty-Fifth AAAI Conference on Artificial Intelligence*. 2011.

Borthwick, Andrew and Ralph Grishman. “A maximum entropy approach to named entity recognition”. PhD thesis. Citeseer, 1999.

Borthwick, Andrew, John Sterling, Eugene Agichtein, and Ralph Grishman. “Exploiting diverse knowledge sources via maximum entropy in named entity recognition”. In: *Sixth Workshop on Very Large Corpora*. 1998.

Bouma, Gerlof. “Normalized (pointwise) mutual information in collocation extraction”. In: *Proceedings of GSCL* (2009), pp. 31–40.

Breschi, Stefano and Franco Malerba. “The geography of innovation and economic clustering: some introductory notes”. In: *Industrial and corporate change* 10.4 (2001), pp. 817–833.

Brickley, Dan, Ramanathan V Guha, and Andrew Layman. “Resource description framework (RDF) schema specification”. In: (1999).

Brin, Sergey. “Extracting patterns and relations from the world wide web”. In: *International Workshop on The World Wide Web and Databases*. Springer. 1998, pp. 172–183.

Brodie, Michael L and John Mylopoulos. “Knowledge bases vs databases”. In: *On Knowledge Base Management Systems*. Springer, 1986, pp. 83–86.

Brown, Theodore L, Harold Eugene LeMay, Bruce Edward Bursten, and Bruce E Bursten. *Chemistry: the central science*. Vol. 8. Prentice Hall Englewood Cliffs, NJ, 1994.

Bruni, Elia, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. “Distributional semantics in technicolor”. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics. 2012, pp. 136–145.

Buchanan, BG, GL Sutherland, and EA Feigenbaum. “Heuristic DENDRAL: A program for generating processes in organic chemistry”. In: *Machine Intelligence* 4 (1969).

Budanitsky, Alexander and Graeme Hirst. “Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures”. In: *Workshop on WordNet and other lexical resources*. Vol. 2. 2001, pp. 2–2.

Budanitsky, Alexander and Graeme Hirst. “Evaluating wordnet-based measures of lexical semantic relatedness”. In: *Computational Linguistics* 32.1 (2006), pp. 13–47.

Buitelaar, Paul and Philipp Cimiano. *Ontology learning and population: bridging the gap between text and knowledge*. Vol. 167. Ios Press, 2008.

Carley, Kathleen. “Knowledge acquisition as a social phenomenon”. In: *Instructional Science* 14.3-4 (1986), pp. 381–438.

Celebi, M Emre, Hassan A Kingravi, and Patricio A Vela. “A comparative study of efficient initialization methods for the k-means clustering algorithm”. In: *Expert systems with applications* 40.1 (2013), pp. 200–210.

Chaffin, Roger and Douglas J Herrmann. “The similarity and diversity of semantic relations”. In: *Memory & Cognition* 12.2 (1984), pp. 134–141.

Chakraborty, Goutam, Mitsuru Murakami, Norio Shiratori, and Shoichi Noguchi. “A growing network that optimizes between undertraining and overtraining”. In: *Proceedings of ICNN’95-International Conference on Neural Networks*. Vol. 2. IEEE. 1995, pp. 1116–1120.

Chakraborty, Sanjay, NK Nagwani, and Lopamudra Dey. “Weather forecasting using incremental K-means clustering”. In: *arXiv preprint arXiv:1406.4756* (2014).

Chen, Yun-Nung, William Yang Wang, and Alexander Rudnicky. “Jointly modeling inter-slot relations by random walk on knowledge graphs for unsupervised spoken language understanding”. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015, pp. 619–629.

Chiu, Jason PC and Eric Nichols. “Named entity recognition with bidirectional LSTM-CNNs”. In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 357–370.

Choi, Eunsol, Omer Levy, Yejin Choi, and Luke Zettlemoyer. “Ultra-Fine Entity Typing”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 87–96.

Chollet, François et al. “Keras documentation”. In: *Keras. io* (2015).

Church, Kenneth Ward and Patrick Hanks. “Word association norms, mutual information, and lexicography”. In: *Computational linguistics* 16.1 (1990), pp. 22–29.

Clark, Eve V. “Conventionality and contrast: Pragmatic principles with lexical consequences”. In: *Lehrer and Kittay, 1992a* (1992), pp. 171–188.

Collins, Michael and Yoram Singer. “Unsupervised models for named entity classification”. In: *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. 1999.

Collobert, Ronan and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 160–167.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch". In: *Journal of machine learning research* 12.ARTICLE (2011), pp. 2493–2537.

Cruse, D Alan, David Alan Cruse, D A Cruse, and D A Cruse. *Lexical semantics*. Cambridge university press, 1986.

Cucchiarelli, Alessandro, Danilo Luzi, and Paola Velardi. "Automatic semantic tagging of unknown proper names". In: *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*. 1998.

Cucchiarelli, Alessandro and Paola Velardi. "Unsupervised named entity recognition using syntactic and semantic contextual evidence". In: *Computational Linguistics* 27.1 (2001), pp. 123–131.

Cui, Leyang and Yue Zhang. "Hierarchically-Refined Label Attention Network for Sequence Labeling". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 4106–4119.

Culotta, Aron, Ron Bekkerman, and Andrew McCallum. *Extracting social networks and contact information from email and the web*. Tech. rep. MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE, 2005.

Culotta, Aron, Andrew McCallum, and Jonathan Betz. "Integrating probabilistic extraction models and data mining to discover relations and patterns in text". In: *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics. 2006, pp. 296–303.

Curran, James. “Supersense Tagging of Unknown Nouns Using Semantic Similarity”. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Ann Arbor, MI, 2005, pp. 26–33. DOI: 10.3115/1219840.1219844. URL: http://www.aclweb.org/anthology/P05-1004.

Daoud, Jamal I. “Multicollinearity and regression analysis”. In: *Journal of Physics: Conference Series*. Vol. 949. 1. IOP Publishing. 2017, p. 012009.

Das, Priyanka, Asit Kumar Das, Janmenjoy Nayak, and Danilo Pelusi. “A framework for crime data analysis using relationship among named entities”. In: *Neural Computing and Applications* (2019), pp. 1–19.

Dasgupta, Sanjoy. *The hardness of k-means clustering*. Department of Computer Science and Engineering, University of California . . ., 2008.

Davidov, Dmitry and Ari Rappoport. “Extraction and approximation of numerical attributes from the Web”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, 2010, pp. 1308–1317.

Davis, Randall. “Interactive transfer of expertise: Acquisition of new inference rules”. In: *Artificial intelligence* 12.2 (1979), pp. 121–157.

Deese, James. “The associative structure of some common English adjectives”. In: *Journal of Verbal Learning and Verbal Behavior* 3.5 (1964), pp. 347–357.

Del Corro, Luciano, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. “Finet: Context-aware fine-grained named entity typing”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 868–878.

Dennett, Daniel C. *Consciousness explained*. Penguin uk, 1993.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, Volume 1 (Long and Short Papers). 2019, pp. 4171–4186.

Dhanachandra, Nameirakpam and Yambem Jina Chanu. "A survey on image segmentation methods using clustering techniques". In: *European Journal of Engineering Research and Science* 2.1 (2017), pp. 15–20.

Dinu, Georgiana and Marco Baroni. "How to make words with vectors: Phrase generation in distributional semantics". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014, pp. 624–633.

Dinu, Georgiana, Nghia The Pham, and Marco Baroni. "DISSECT: DISTributional SEmantics Composition Toolkit". In: *Proceedings of ACL (System Demonstrations)*. Sofia, Bulgaria, 2013, pp. 31–36.

Doddington, George R, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. "The automatic content extraction (ace) program-tasks, data, and evaluation." In: *Lrec*. Vol. 2. 1. Lisbon. 2004, pp. 837–840.

Dubes, Richard C. "How many clusters are best?-an experiment". In: *Pattern Recognition* 20.6 (1987), pp. 645–663.

Efron, Bradley and Robert Tibshirani. *An Introduction to the Bootstrap*. Boca Raton, FL: Chapman and Hall, 1994.

Erk, Katrin. "What is word meaning, really?:(and how can distributional models help us describe it?)" In: *Proceedings of the 2010 workshop on geometrical models of natural language semantics*. Association for Computational Linguistics. 2010, pp. 17–26.

Erk, Katrin. "Vector space models of word meaning and phrase meaning: A survey". In: *Language and Linguistics Compass* 6.10 (2012), pp. 635–653.

Erk, Katrin, Sebastian Padó, and Ulrike Padó. "A flexible, corpus-driven model of regular and inverse selectional preferences". In: *Computational Linguistics* 36.4 (2010), pp. 723–763.

Ethayarajh, Kawin. "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2

Embeddings". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 55–65.

Etzioni, Oren, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. "Unsupervised named-entity extraction from the web: An experimental study". In: *Artificial intelligence* 165.1 (2005), pp. 91–134.

Evans, Richard and Stafford Street. "A framework for named entity recognition in the open domain". In: *Recent advances in natural language processing III: selected papers from RANLP* 260.267-274 (2003), p. 110.

Everitt, Brian S, Sabine Landau, Morven Leese, and Daniel Stahl. *Cluster analysis*. John Wiley & Sons, 2011.

Evert, Stefan. "The Statistics of Word Cooccurrences". Ph.D dissertation. Stuttgart University, 2005.

Făgărăşan, Luana, Eva Maria Vecchi, and Stephen Clark. "From distributional semantics to feature norms: grounding semantic models in human perceptual data". In: *Proceedings of IWCS*. London, UK, 2015, pp. 52–57.

Farrar, Donald E and Robert R Glauber. "Multicollinearity in regression analysis: the problem revisited". In: *The Review of Economic and Statistics* (1967), pp. 92–107.

Faruqui, Manaal and Chris Dyer. "Non-distributional Word Vector Representations". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 2015, pp. 464–469.

Feigenbaum, Edward A. *The art of artificial intelligence. 1. Themes and case studies of knowledge engineering*. Tech. rep. Stanford Univ CA Dept of Computer Science, 1977.

Fellbaum, Christiane, ed. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.

Firth, J. R. “A synopsis of linguistic theory 1930–55”. In: *Studies in Linguistic Analysis*. Oxford: The Philological Society, 1957, pp. 1–32.

Fisher, Joseph and Andreas Vlachos. “Merge and Label: A Novel Neural Network Architecture for Nested NER”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 5840–5850.

Fleischman, Michael. “Automated subcategorization of named entities”. In: *ACL (Companion Volume)*. 2001, pp. 25–30.

Fodor, Jerry and Ernie Lepore. “All at sea in semantic space: Churchland on meaning similarity”. In: *the Journal of Philosophy* 96.8 (1999), pp. 381–403.

Freitas, Andre and Edward Curry. “Natural language queries over heterogeneous linked data graphs: A distributional-compositional semantics approach”. In: *Proceedings of the 19th international conference on Intelligent User Interfaces*. ACM. 2014, pp. 279–288.

Freitas, André, Edward Curry, Joao Gabriel Oliveira, and Sean O’Riain. “Querying heterogeneous datasets on the linked data web: challenges, approaches, and trends”. In: *IEEE Internet Computing* 16.1 (2012), pp. 24–33.

Freitas, André, Joao Carlos Pereira da Silva, Edward Curry, and Paul Buitelaar. “A distributional semantics approach for selective reasoning on commonsense graph knowledge bases”. In: *International Conference on Applications of Natural Language to Data Bases/Information Systems*. Salford, UK, 2014, pp. 21–32.

Friedman, Alinda, Dennis Kerkman, and Norman Brown. “Spatial location judgments: A cross-national comparison of estimation bias in subjective North American geography”. In: *Psychonomic Bulletin & Review* 9.3 (2002), pp. 615–623.

Frome, Andrea, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc’Aurelio Ranzato, and Tomas Mikolov. “Devise: A deep visual-semantic embedding model”. In: *Advances in neural information processing systems*. 2013, pp. 2121–2129.

Fu, Ruiji, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. "Learning semantic hierarchies via word embeddings". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014, pp. 1199–1209.

Fujiwara, Yasuhiro and Go Irie. "Efficient label propagation". In: *International conference on machine learning*. PMLR. 2014, pp. 784–792.

Fukunaga, Keinosuke. *Introduction to statistical pattern recognition*. Elsevier, 2013.

Gan, Haitao, Rui Huang, Zhizeng Luo, Xugang Xi, and Yunyuan Gao. "On using supervised clustering analysis to improve classification performance". In: *Information Sciences* 454 (2018), pp. 216–228.

Gan, Haitao, Nong Sang, Rui Huang, Xiaojun Tong, and Zhiping Dan. "Using clustering analysis to improve semi-supervised classification". In: *Neurocomputing* 101 (2013), pp. 290–298.

Gangemi, Aldo, Nicola Guarino, and Alessandro Oltramari. "Conceptual analysis of lexical taxonomies: The case of WordNet top-level". In: *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*. 2001, pp. 285–296.

Gao, Xinbo, Bing Xiao, Dacheng Tao, and Xuelong Li. "A survey of graph edit distance". In: *Pattern Analysis and applications* 13.1 (2010), pp. 113–129.

Gärdenfors, Peter. *Conceptual spaces: The geometry of thought*. MIT press, 2004.

Garg, Dinesh, Shajith Ikbal, Santosh K Srivastava, Harit Vishwakarma, Hima Karanam, and L Venkata Subramaniam. "Quantum embedding of knowledge for reasoning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 5594–5604.

Geeraerts, Dirk. *Theories of Lexical Semantics*. Oxford University Press, 2010. ISBN: 978-0198700319.

Geffet, Maayan and Ido Dagan. "The distributional inclusion hypotheses and lexical entailment". In: *Proceedings of the 43rd Annual Meeting on*

Association for Computational Linguistics. Association for Computational Linguistics. 2005, pp. 107–114.

Ghaddar, Abbas and Philippe Langlais. "Robust Lexical Features for Improved Neural Network Named-Entity Recognition". In: *Proceedings of the 27th International Conference on Computational Linguistics*. 2018, pp. 1896–1907.

Ghamrawi, Nadia and Andrew McCallum. "Collective multi-label classification". In: *Proceedings of the 14th ACM international conference on Information and knowledge management*. 2005, pp. 195–200.

Ghannay, Sahar, Benoit Favre, Yannick Esteve, and Nathalie Camelin. "Word embedding evaluation and combination". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 2016, pp. 300–305.

Gillick, Dan, Nevena Lazic, Kuzman Ganchev, Jessica Kirchner, and David Huynh. "Context-Dependent Fine-Grained Entity Type Tagging". In: (2014).

Glonek, Garique FV and Peter McCullagh. "Multivariate logistic models". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 57.3 (1995), pp. 533–546.

Goldfarb, Charles F. *The SGML handbook*. Oxford University Press, 1990.

Grefenstette, Gregory. *Corpus-derived First, Second, and Third-order Word Affinities*. Rank Xerox Research Centre, 1994.

Grishman, Ralph and Beth M Sundheim. "Message understanding conference-6: A brief history". In: *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. 1996.

Guo, Yuanbo, Zhengxiang Pan, and Jeff Heflin. "LUBM: A benchmark for OWL knowledge base systems". In: *Journal of Web Semantics* 3.2-3 (2005), pp. 158–182.

Guu, Kelvin, John Miller, and Percy Liang. "Traversing knowledge graphs in vector space". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal, 2015, pp. 318–327.

Haarslev, Volker and Ralf Möller. “Racer: A Core Inference Engine for the Semantic Web.” In: *EON*. Vol. 87. 2003.

Hamon, Thierry and Adeline Nazarenko. “Detection of synonymy links between terms: experiment and results”. In: *Recent advances in computational terminology* 2 (2001), pp. 185–208.

Hampton, James A. “Polymorphous concepts in semantic memory”. In: *Journal of verbal learning and verbal behavior* 18.4 (1979), pp. 441–461.

Harris, Zellig S. “Distributional structure”. In: *Word* 10.2-3 (1954), pp. 146–162.

Hearst, Marti A. “Automatic acquisition of hyponyms from large text corpora”. In: *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics. 1992, pp. 539–545.

Hebboul, Amel, Fella Hachouf, and Amel Boulemnadjel. “A new incremental neural network for simultaneous clustering and classification”. In: *Neurocomputing* 169 (2015), pp. 89–99.

Heim, Irene and Angelika Kratzer. *Semantics in Generative Grammar*. Malden, MA: Blackwell, 1998.

Hendrickx, Iris, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. “SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals”. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. 2010, pp. 33–38.

Herbelot, Aurélie. “Mr Darcy and Mr Toad, Gentlemen: Distributional names and their kinds”. In: *Proceedings of the International Conference on Computational Semantics*. Berlin, Germany, 2015, pp. 151–161.

Herbelot, Aurélie and Marco Baroni. “High-risk learning: acquiring new word vectors from tiny data”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 304–309.

Herbelot, Aurélie and Eva Maria Vecchi. "Building a shared world: mapping distributional to model-theoretic semantic spaces". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal, 2015, pp. 22–32. URL: http://aclweb.org/anthology/D15-1003.

Hill, Felix, Roi Reichart, and Anna Korhonen. "Multi-modal models for concrete and abstract concept meaning". In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 285–296.

Hopper, Paul J and Sandra A Thompson. "The discourse basis for lexical categories in universal grammar". In: *Language* 60.4 (1984), pp. 703–752.

Hovy, Eduard, Roberto Navigli, and Simone Paolo Ponzetto. "Collaboratively Built Semi-structured Content and Artificial Intelligence: The Story So Far". In: *Artificial Intelligence* 194 (Jan. 2013), pp. 2–27. ISSN: 0004-3702. DOI: 10.1016/j.artint.2012.10.002. URL: http://dx.doi.org/10.1016/j.artint.2012.10.002.

Huang, Zhiheng, Marcus Thint, and Zengchang Qin. "Question classification using head words and their hypernyms". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2008, pp. 927–936.

Iwakura, Tomoya, Kanako Komiya, and Ryuichi Tachibana. "Constructing a Japanese basic named entity corpus of various genres". In: *Proceedings of the Sixth Named Entity Workshop*. 2016, pp. 41–46.

Jain, Anil K and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

Jain, Anil K, M Narasimha Murty, and Patrick J Flynn. "Data clustering: a review". In: *ACM computing surveys (CSUR)* 31.3 (1999), pp. 264–323.

Jarrar, Mustafa and Robert Meersman. "Scalability and knowledge reusability in ontology modeling". In: Scuola Superiore G Reiss Romoli. 2002.

Jebbara, Soufian, Valerio Basile, Elena Cabrio, and Philipp Cimiano. "Extracting common sense knowledge via triple ranking using supervised

and unsupervised distributional models". In: *Semantic Web* 10.1 (2019), pp. 139–158.

Ji, Heng and Ralph Grishman. "Data selection in semi-supervised learning for name tagging". In: *Proceedings of the Workshop on Information Extraction Beyond The Document*. 2006, pp. 48–55.

Jie, Zhanming and Wei Lu. "Dependency-Guided LSTM-CRF for Named Entity Recognition". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 3853–3863.

Johns, Brendan and Michael Jones. "Perceptual Inference Through Global Lexical Similarity". In: *Topics in Cognitive Science* 4.1 (2012), pp. 103–120.

Johnson, Stephen C. "Hierarchical clustering schemes". In: *Psychometrika* 32.3 (1967), pp. 241–254.

Jolliffe, Ian. *Principal component analysis*. Springer, 2011.

Jones, Steven. *Antonymy: a corpus-based perspective*. Routledge, 2003.

Justeson, John S and Slava M Katz. "Redefining Antonymy: The Textual Structure of a Semantic Relation1". In: *Literary and Linguistic Computing* 7.3 (1992), pp. 176–184.

Kartsaklis, Dimitri, Mehrnoosh Sadrzadeh, and Stephen Pulman. "Separating disambiguation from composition in distributional semantics". In: *Proceedings of CoNLL*. Sofia, Bulgaria, 2013, pp. 114–123.

Katz, Jerrold J and Jerry A Fodor. "The structure of a semantic theory". In: *language* 39.2 (1963), pp. 170–210.

Kaufmann, Leonard and Peter J Rousseeuw. "Finding groups in data: an introduction to cluster analysis". In: *New York: Jonh Wiley* (1990).

Kern, Willis F. and James R. Bland. *Solid Mensuration with Proofs*. 2nd. New York: Wiley, 1948.

Kesavaraj, Gopalan and Sreekumar Sukumaran. "A study on classification techniques in data mining". In: *2013 fourth international conference on*

computing, communications and networking technologies (ICCCNT). IEEE. 2013, pp. 1–7.

Khot, Tushar, Ashish Sabharwal, and Peter Clark. "Scitail: A textual entailment dataset from science question answering". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

Kiela, Douwe and Stephen Clark. "A systematic study of semantic vector space model parameters". In: *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*. 2014, pp. 21–30.

Kim, Bosung, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. "Multi-task learning for knowledge graph completion with pre-trained language models". In: *Proceedings of the 28th International Conference on Computational Linguistics*. 2020, pp. 1737–1743.

Klein, Dan and Christopher D Manning. "Accurate unlexicalized parsing". In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics. 2003, pp. 423–430.

Klein, Ewan. "A semantics for positive and comparative adjectives". In: *Linguistics and philosophy* 4.1 (1980), pp. 1–45.

Kleinberg, Jon M. "An impossibility theorem for clustering". In: *Advances in neural information processing systems*. 2003, pp. 463–470.

Kononenko, Igor. "Automatic Knowledge Acquisition". In: *Current trends in knowledge acquisition* 8 (1990), p. 190.

Kotnis, Bhushan and Alberto García-Durán. "Learning Numeric Attributes in Knowledge Bases". In: *Proceedings of AKBC*. Amherst, MA, 2019.

Kramer, Oliver. "K-nearest neighbors". In: *Dimensionality reduction with unsupervised nearest neighbors*. Springer, 2013, pp. 13–23.

Kripke, Saul A. *Wittgenstein on rules and private language: An elementary exposition*. Harvard University Press, 1982.

Krishnamurthy, Jayant and Tom M Mitchell. “Learning a compositional semantics for Freebase with an open predicate vocabulary”. In: *Transactions of the Association for Computational Linguistics* 3 (2015), pp. 257–270.

Krishnapuram, Balaji, David Williams, Ya Xue, Lawrence Carin, Mário Figueiredo, and Alexander Hartemink. “On semi-supervised classification”. In: *Advances in neural information processing systems* 17 (2004).

Krupka, George. “SRA: Description of the SRA System as Used for MUC-6”. In: *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*. 1995.

Krupka, George and Kevin Hausman. “IsoQuest Inc.: Description of the NetOwl™ Extractor System as Used for MUC-7”. In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*. 1998.

Kruszewski, Germán, Denis Paperno, and Marco Baroni. “Deriving Boolean structures from distributional vectors”. In: *Transactions of the Association for Computational Linguistics* 3 (2015), pp. 375–388.

Lakoff, Robin. “Language and woman’s place”. In: *Language in society* 2.1 (1973), pp. 45–79.

Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. “Neural Architectures for Named Entity Recognition”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, pp. 260–270.

Landauer, Thomas K and Susan T Dumais. “A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge”. In: *Psychological review* 104.2 (1997), p. 211.

Lapesa, Gabriella and Stefan Evert. “A large scale evaluation of distributional semantic models: Parameters, interactions and model selection”.

In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 531–546.

Larochelle, Hugo, Yoshua Bengio, et al. *Distributed representation prediction for generalization to new words*. Tech. rep. 2006.

Lazaridou, Angeliki, Marco Baroni, et al. "A multitask objective to inject lexical contrast into distributional semantics". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 2015, pp. 21–26.

Lazaridou, Angeliki, Elia Bruni, and Marco Baroni. "Is this a wampimuk? Cross-modal mapping between distributional semantics and the visual world". In: *Proceedings of ACL*. Baltimore, MD, 2014, pp. 1403–1414.

Le, Quoc and Tomas Mikolov. "Distributed representations of sentences and documents". In: *International conference on machine learning*. 2014, pp. 1188–1196.

Lee, Lillian. "Measures of Distributional Similarity". In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. 1999, pp. 25–32.

Lenci, Alessandro. "Distributional semantics in linguistic and cognitive research". In: *Italian journal of linguistics* 20.1 (2008), pp. 1–31.

Lenci, Alessandro. "Distributional models of word meaning". In: *Annual review of Linguistics* 4 (2018), pp. 151–171.

Lenci, Alessandro and Giulia Benotto. "Identifying hypernyms in distributional semantic spaces". In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics*. Montreal, Canada, 2012, pp. 75–79.

Lenci, Alessandro, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. "A comparative evaluation and analysis of three generations of Distributional Semantic Models". In: *Language Resources and Evaluation* (2022), pp. 1–45.

Leser, Ulf and Jörg Hakenberg. “What makes a gene name? Named entity recognition in the biomedical literature”. In: *Briefings in bioinformatics* 6.4 (2005), pp. 357–369.

Levy, Omer and Yoav Goldberg. “Dependency-based word embeddings”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2014, pp. 302–308.

Levy, Omer and Yoav Goldberg. “Linguistic regularities in sparse and explicit word representations”. In: *Proceedings of CoNLL*. Ann Arbor, MI, 2014, pp. 171–180.

Levy, Omer, Yoav Goldberg, and Ido Dagan. “Improving distributional similarity with lessons learned from word embeddings”. In: *Transactions of the Association for Computational Linguistics* 3 (2015), pp. 211–225.

Levy, Omer, Steffen Remus, Chris Biemann, and Ido Dagan. “Do supervised distributional methods really learn lexical inference relations?” In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015, pp. 970–976.

Lewis, Mike and Mark Steedman. “Combined distributional and logical semantics”. In: *Transactions of the Association for Computational Linguistics* 1 (2013), pp. 179–192.

Li, Peng-Hsuan, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun Ma. “Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 2664–2669.

Liang, Percy. “Semi-supervised learning for natural language”. PhD thesis. Massachusetts Institute of Technology, 2005.

Limaye, Girija, Sunita Sarawagi, and Soumen Chakrabarti. “Annotating and searching web tables using entities, types and relationships”. In: *Proceedings of the VLDB Endowment* 3.1-2 (2010), pp. 1338–1347.

Lin, Dekang and Xiaoyun Wu. "Phrase clustering for discriminative learning". In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. 2009, pp. 1030–1038.

Lin, Yankai, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. "Learning entity and relation embeddings for knowledge graph completion." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Austin, TX, 2015, pp. 2181–2187.

Lin, Yung-Shen, Jung-Yi Jiang, and Shie-Jue Lee. "A similarity measure for text classification and clustering". In: *IEEE transactions on knowledge and data engineering* 26.7 (2013), pp. 1575–1590.

Ling, Xiao and Daniel S Weld. "Fine-grained entity recognition". In: *Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012.

Ling, Xiao and Daniel S. Weld. "Fine-grained Entity Recognition". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Toronto Canada, 2012, pp. 94–100.

Lloyd, Stuart. "Least squares quantization in PCM". In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

Lopez, Vanessa, Christina Unger, Philipp Cimiano, and Enrico Motta. "Evaluating question answering over linked data". In: *Journal of Web Semantics* 21 (2013), pp. 3–13.

Louwerse, Max and Rolf Zwaan. "Language encodes geographical information". In: *Cognitive Science* 33 (2009), pp. 51–73.

Lowe, Will. "Towards a theory of semantic space". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 23. 23. 2001.

Lu, Di, Leonardo Neves, Vitor Carvalho, Ning Zhang, and Heng Ji. "Visual attention model for name tagging in multimodal social media". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 1990–1999.

Lucas, Peter and Linda Van Der Gaag. *Principles of expert systems*. Addison-Wesley Wokingham, 1991.

Lund, Kevin and Curt Burgess. “Hyperspace analog to language (HAL): A general model of semantic representation”. In: (1996).

Lund, Kevin and Curt Burgess. “Producing high-dimensional semantic spaces from lexical co-occurrence”. In: *Behavior Research Methods* 28 (1996), pp. 203–208.

Ma, Xuezhe and Eduard Hovy. “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 1064–1074.

Ma, Yukun, Erik Cambria, and Sa Gao. “Label embedding for zero-shot fine-grained named entity typing”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016, pp. 171–180.

MacGregor, Robert and David Brill. “Recognition algorithms for the LOOM classi er”. In: *Proc. of the 10th Nat. Conf. on Artificial Intelligence AAAI-92*. 1992, p. 774.

MacGregor, Robert and Mark H Burstein. “Using a description classifier to enhance knowledge representation”. In: *IEEE Expert* 6.3 (1991), pp. 41–46.

MacQueen, James et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

Mai, Khai, Thai-Hoang Pham, Minh Trung Nguyen, Tuan Duc Nguyen, Danushka Bollegala, Ryohei Sasano, and Satoshi Sekine. “An empirical study on fine-grained named entity recognition”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. 2018, pp. 711–722.

Manning, Christopher D, Christopher D Manning, and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

Martinet, André. “Reflexions sur la signification”. In: *La linguistique* 25.Fasc. 1 (1989), pp. 43–51.

McCallum, Andrew and Wei Li. “Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons”. In: (2003).

McCarthy, Diana and John Carroll. “Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences”. In: *Computational Linguistics* 29.4 (2003), pp. 639–654.

McCarthy, Diana and Roberto Navigli. “The English Lexical Substitution Task”. In: *Language Resources and Evaluation* 43.2 (2009), pp. 139–159.

McNally, Louise and Gemma Boleda. “Conceptual vs. Referential Affordance in Concept Composition”. In: *Compositionality and Concepts in Linguistics and Psychology*. Ed. by Yoad Winter and James Hampton. Springer, 2017.

Medin, Douglas L and Marguerite M Schaffer. “Context theory of classification learning.” In: *Psychological review* 85.3 (1978), p. 207.

Menon, Aditya Krishna and Robert C Williamson. “The cost of fairness in binary classification”. In: *Conference on Fairness, Accountability and Transparency*. PMLR. 2018, pp. 107–118.

Mikheev, Andrei, Marc Moens, and Claire Grover. “Named entity recognition without gazetteers”. In: *Ninth Conference of the European Chapter of the Association for Computational Linguistics*. 1999.

Mikolov, Tomáš. “Language modeling for speech recognition in czech”. In: *Ph. D. dissertation, Masters thesis* (2007).

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

Mikolov, Tomas, Jiri Kopecky, Lukas Burget, Ondrej Glembek, et al. “Neural network based language models for highly inflective languages”. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2009, pp. 4725–4728.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic regularities in continuous space word representations". In: *Proceedings of NAACL*. Atlanta, Georgia, 2013, pp. 746–751.

Miller, George A. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

Miller, George A and Florentina Hristea. "WordNet nouns: Classes and instances". In: *Computational linguistics* 32.1 (2006), pp. 1–3.

Min, Bonan, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. "Distant supervision for relation extraction with an incomplete knowledge base". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia, 2013, pp. 777–782.

Mitchell, Alexis, Stephanie Strassel, Mark Przybocki, JK Davis, George Doddington, Ralph Grishman, Adam Meyers, Ada Brunstein, Lisa Ferro, and Beth Sundheim. "ACE-2 Version 1.0". In: *Linguistic Data Consortium, Philadelphia* (2003).

Mitchell, Jeff and Mirella Lapata. "Composition in distributional models of semantics". In: *Cognitive Science* 34.8 (2010), pp. 1388–1429.

Mitchell, Tom, Svetlana Shinkareva, Andrew Carlson, Kai-Min Chang, Vincente Malave, Robert Mason, and Marcel Just. "Predicting human brain activity associated with the meanings of nouns". In: *Science* 320 (2008), pp. 1191–1195.

Mnih, Andriy and Geoffrey E Hinton. "A scalable hierarchical distributed language model". In: *Advances in neural information processing systems*. 2009, pp. 1081–1088.

Moen, SPFGH and Tapio Salakoski2 Sophia Ananiadou. "Distributional semantics resources for biomedical text processing". In: *Proceedings of LBM* (2013), pp. 39–44.

Mohammad, Saif M and Graeme Hirst. “Distributional measures of semantic distance: A survey”. In: *arXiv preprint arXiv:1203.1858* (2012).

Montague, Richard. “Universal Grammar”. In: *Theoria* 36 (1970), pp. 373–398.

Morin, Frederic and Yoshua Bengio. “Hierarchical probabilistic neural network language model.” In: *Aistats*. Vol. 5. Citeseer. 2005, pp. 246–252.

Morris, Jane and Graeme Hirst. “Non-classical lexical semantic relations”. In: *Proceedings of the Computational Lexical Semantics Workshop at HLT-NAACL 2004*. 2004, pp. 46–51.

Murphy, Brian, Partha Talukdar, and Tom Mitchell. “Learning effective and interpretable semantic models using non-negative sparse embedding”. In: *Proceedings of COLING 2012*. 2012, pp. 1933–1950.

Murphy, Brian, Partha Talukdar, and Tom Mitchell. “Selecting corpus-semantic models for neurolinguistic decoding”. In: *Proceedings of *SEM*. Montreal, Canada, 2012, pp. 114–123.

Murphy, Gregory. *The big book of concepts*. MIT press, 2004.

Murphy, Kevin P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Nadeau, David and Satoshi Sekine. “A survey of named entity recognition and classification”. In: *Lingvisticae Investigationes* 30.1 (2007), pp. 3–26.

Nadkarni, Rahul, David Wadden, Iz Beltagy, Noah Smith, Hannaneh Hajishirzi, and Tom Hope. “Scientific Language Models for Biomedical Knowledge Base Completion: An Empirical Study”. In: *NeurIPS 2021 AI for Science Workshop*. 2021.

Nagy, George. “State of the art in pattern recognition”. In: *Proceedings of the IEEE* 56.5 (1968), pp. 836–863.

Nahm, Un Yong and Raymond J Mooney. “A mutually beneficial integration of data mining and information extraction”. In: *AAAI/IAAI*. 2000, pp. 627–632.

Nakashole, Ndapandula, Martin Theobald, and Gerhard Weikum. “Scalable knowledge harvesting with high precision and high recall”. In: *Proceedings of the fourth ACM international conference on Web search and data mining*. 2011, pp. 227–236.

Narisawa, Katsuma, Yotaro Watanabe, Junta Mizuno, Naoaki Okazaki, and Kentaro Inui. “Is a 204 cm man tall or small? acquisition of numerical common sense from the web”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2013, pp. 382–391.

Neelakantan, Arvind, Benjamin Roth, and Andrew McCallum. “Compositional Vector Space Models for Knowledge Base Completion”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2015, pp. 156–166.

Newell, A and JC Shaw. “A variety op intelligent learning in a general problem solver”. In: *RAND Report P-1742, dated July* 6 (1959).

Nguyen, Kim Anh, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. “Hierarchical Embeddings for Hypernymy Detection and Directionality”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 233–243.

Nickel, Maximilian, Lorenzo Rosasco, and Tomaso A Poggio. “Holographic Embeddings of Knowledge Graphs.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Phoenix, AZ, 2016, pp. 1955–1961.

Niwa, Yoshiki and Yoshihiko Nitta. “Co-occurrence vectors from corpora vs. distance vectors from dictionaries”. In: *Proceedings of the 15th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics. 1994, pp. 304–309.

Noble, William S. “What is a support vector machine?” In: *Nature biotechnology* 24.12 (2006), pp. 1565–1567.

Obeidat, Rasha, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. "Description-based zero-shot fine-grained entity typing". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 807–814.

Oseledets, Ivan V and Eugene E Tyrtyshnikov. "Breaking the curse of dimensionality, or how to use SVD in many dimensions". In: *SIAM Journal on Scientific Computing* 31.5 (2009), pp. 3744–3759.

Ostrovsky, Rafail, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. "The effectiveness of Lloyd-type methods for the k-means problem". In: *Journal of the ACM (JACM)* 59.6 (2013), pp. 1–22.

Padó, Sebastian and Mirella Lapata. "Dependency-based construction of semantic space models". In: *Computational Linguistics* 33.2 (2007), pp. 161–199.

Palatucci, Mark, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. "Zero-shot learning with semantic output codes". In: *Advances in neural information processing systems*. 2009, pp. 1410–1418.

Palmer, David D and David Day. "A statistical profile of the named entity task". In: *Fifth Conference on Applied Natural Language Processing*. 1997, pp. 190–193.

Pantel, Patrick and Marco Pennacchiotti. "Automatically Harvesting and Ontologizing Semantic Relations". In: *Ontology Learning and Population: Bridging the Gap between Text and Knowledge* (2008), p. 171.

Pasca, Marius, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. "Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge". In: *AAAI*. Vol. 6. 2006, pp. 1400–1405.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. “Deepwalk: Online learning of social representations”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. New York City, NY, 2014, pp. 701–710.

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2018, pp. 2227–2237.

Plank, Barbara and Alessandro Moschitti. “Embedding semantic similarity in tree kernels for domain adaptation of relation extraction”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2013, pp. 1498–1507.

Quirk, Randolph. *A comprehensive grammar of the English language*. Pearson Education India, 2010.

Radovanović, Miloš, Alexandros Nanopoulos, and Mirjana Ivanović. “Hubs in space: Popular nearest neighbors in high-dimensional data”. In: *Journal of Machine Learning Research* 11 (2010), pp. 2487–2531.

Rau, Lisa F. “Extracting company names from text”. In: *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*. Vol. 1. IEEE. 1991, pp. 29–32.

Rebele, Thomas, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. “YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames”. In: *International Semantic Web Conference*. Springer. 2016, pp. 177–185.

Reed, Stephen K. “Pattern recognition and categorization”. In: *Cognitive psychology* 3.3 (1972), pp. 382–407.

Reiter, Nils and Anette Frank. “Identifying generic noun phrases”. In: *Proceedings of the 48th annual meeting of the association for computational linguistics*. 2010, pp. 40–49.

Resnik, Philip. “Selectional constraints: An information-theoretic model and its computational realization”. In: *Cognition* 61.1-2 (1996), pp. 127–159.

Rich, Steven H and V Venkatasubramanian. “Model-based reasoning in diagnostic expert systems for chemical process plants”. In: *Computers & Chemical Engineering* 11.2 (1987), pp. 111–122.

Rigau, German, Jordi Atserias, and Eneko Agirre. “Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation”. In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*. Madrid, Spain, 1997, pp. 48–55. DOI: 10.3115/976909.979624. URL: http://www.aclweb.org/anthology/P97-1007.

Rocktäschel, Tim, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. “Reasoning about entailment with neural attention”. In: *arXiv preprint arXiv:1509.06664* (2015).

Roller, Stephen and Katrin Erk. “Relations such as Hypernymy: Identifying and Exploiting Hearst Patterns in Distributional Vectors for Lexical Entailment”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas, 2016, pp. 2163–2172. URL: https://aclweb.org/anthology/D16-1234.

Roller, Stephen, Katrin Erk, and Gemma Boleda. “Inclusive yet selective: Supervised distributional hypernymy detection”. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 2014, pp. 1025–1036.

Rosch, Eleanor. “Cognitive representations of semantic categories”. In: *Journal of Experimental Psychology: General* 104 (1975), pp. 192–233.

Rosch, Eleanor and Carolyn B Mervis. “Family resemblances: Studies in the internal structure of categories”. In: *Cognitive psychology* 7.4 (1975), pp. 573–605.

Rosch, Eleanor, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem. “Basic objects in natural categories”. In: *Cognitive psychology* 8.3 (1976), pp. 382–439.

Rosset, Saharon, Claudia Perlich, and Bianca Zadrozny. “Ranking-based evaluation of regression models”. In: *Fifth IEEE International Conference on Data Mining (ICDM’05)*. IEEE. 2005, 8–pp.

Roth, Michael and Kristian Woodsend. “Composition of word representations improves semantic role labelling”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 407–413.

Rubinstein, Dana, Effi Levi, Roy Schwartz, and Ari Rappoport. “How well do distributional models capture different types of semantic knowledge?” In: *Proceedings of ACL (Volume 2: Short Papers)*. Beijing, China, 2015, pp. 726–730.

Ruiz-Casado, Maria, Enrique Alfonseca, Manabu Okumura, and Pablo Castells. “Information Extraction and Semantic Annotation of Wikipedia”. In: *Ontology Learning and Population: Bridging the Gap between Text and Knowledge* (2008), p. 145.

Ruspini, Enrique H. “A new approach to clustering”. In: *Information and control* 15.1 (1969), pp. 22–32.

Sadrzadeh, Mehrnoosh, Dimitri Kartsaklis, and Esma Balkır. “Sentence entailment in compositional distributional semantics”. In: *Annals of Mathematics and Artificial Intelligence* 82.4 (2018), pp. 189–218.

Sahlgren, Magnus. “The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces”. PhD thesis. Stockholm University, 2006.

Sahlgren, Magnus. “The distributional hypothesis”. In: *Italian Journal of Disability Studies* 20 (2008), pp. 33–53.

Sandhaus, Evan. “The new york times annotated corpus”. In: *Linguistic Data Consortium, Philadelphia* 6.12 (2008), e26752.

Santafe, Guzman, Iñaki Inza, and Jose A Lozano. “Dealing with the evaluation of supervised classification algorithms”. In: *Artificial Intelligence Review* 44.4 (2015), pp. 467–508.

Santus, Enrico, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. “Nine features in a random forest to learn taxonomical semantic relations”. In: *arXiv preprint arXiv:1603.08702* (2016).

Santus, Enrico, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. “Chasing hypernyms in vector spaces with entropy”. In: *Proceedings of EACL*. Gothenburg, Sweden, 2014, pp. 38–42.

Santus, Enrico, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. “Evalution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models”. In: *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*. 2015, pp. 64–69.

Šarić, Frane, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. “Takelab: Systems for measuring semantic text similarity”. In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics. 2012, pp. 441–448.

Sassoon, Galit Weidman. *Vagueness, gradability and typicality: The interpretation of adjectives and nouns*. Brill, 2013.

Schaerf, Andrea. “Reasoning with individuals in concept languages”. In: *Data & Knowledge Engineering* 13.2 (1994), pp. 141–176.

Schnabel, Tobias, Igor Labutov, David Mimno, and Thorsten Joachims. “Evaluation methods for unsupervised word embeddings”. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2015, pp. 298–307.

Schütze, Hinrich. “Word space”. In: *Advances in neural information processing systems*. 1993, pp. 895–902.

Schütze, Hinrich and Jan O Pedersen. “A cooccurrence-based thesaurus and two applications to information retrieval”. In: *Information Processing & Management* 33.3 (1997), pp. 307–318.

Seitner, Julian, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. “A Large DataBase of Hypernymy Relations Extracted from the Web.” In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. 2016, pp. 360–367.

Sekine, Satoshi and Hitoshi Isahara. “IREX: IR & IE Evaluation Project in Japanese.” In: *LREC*. Citeseer. 2000, pp. 1977–1980.

Sekine, Satoshi and Chikashi Nobata. “Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy.” In: *LREC*. Lisbon, Portugal. 2004.

Sekine, Satoshi, Kiyoshi Sudo, and Chikashi Nobata. “Extended Named Entity Hierarchy.” In: *LREC*. 2002.

Seret, Alex, Thomas Verbraken, and Bart Baesens. “A new knowledge-based constrained clustering approach: Theory and application in direct marketing”. In: *Applied Soft Computing* 24 (2014), pp. 316–327.

Shen, Wei, Jianyong Wang, Ping Luo, and Min Wang. “Linden: linking named entities with knowledge base via semantic knowledge”. In: *Proceedings of the 21st international conference on World Wide Web*. ACM. 2012, pp. 449–458.

Shen, Yanyao, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. “Deep Active Learning for Named Entity Recognition”. In: *Proceedings of the 2nd Workshop on Representation Learning for NLP*. 2017, pp. 252–256.

Shimaoka, Sonse, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. “Neural Architectures for Fine-grained Entity Type Classification”. In: *Proceedings of the 15th Conference of the European Chapter of the*

Association for Computational Linguistics: Volume 1, Long Papers. 2017, pp. 1271–1280.

Shortliffe, Edward Hance. *MYCIN: a rule-based computer program for advising physicians regarding antimicrobial therapy selection.* Tech. rep. STANFORD UNIV CALIF DEPT OF COMPUTER SCIENCE, 1974.

Shwartz, Vered and Ido Dagan. "Path-based vs. Distributional Information in Recognizing Lexical Semantic Relations". In: *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*. 2016, pp. 24–29.

Shwartz, Vered, Yoav Goldberg, and Ido Dagan. "Improving Hypernymy Detection with an Integrated Path-based and Distributional Method". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany, 2016, pp. 2389–2398.

Shwartz, Vered, Enrico Santus, and Dominik Schlechtweg. "Hypernyms under Siege: Linguistically-motivated Artillery for Hypernymy Detection". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. 2017, pp. 65–75.

Siegel, Sidney. "Nonparametric statistics for the behavioral sciences." In: (1956).

Sienčnik, Scharolta Katharina. "Adapting word2vec to named entity recognition". In: *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*. 2015, pp. 239–243.

Sikos, Jennifer and Sebastian Padó. "Frame identification as categorization: Exemplars vs prototypes in embeddingland". In: *Proceedings of the 13th International Conference on Computational Semantics-Long Papers*. 2019, pp. 295–306.

Socher, Richard, Danqi Chen, Christopher D Manning, and Andrew Ng. "Reasoning with neural tensor networks for knowledge base completion". In: *Advances in neural information processing systems*. 2013, pp. 926–934.

Socher, Richard, Milind Ganjoo, Christopher D Manning, and Andrew Ng. “Zero-shot learning through cross-modal transfer”. In: *Advances in neural information processing systems*. 2013, pp. 935–943.

Sokolova, Marina, Nathalie Japkowicz, and Stan Szpakowicz. “Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation”. In: *Australasian joint conference on artificial intelligence*. Springer. 2006, pp. 1015–1021.

Speel, Piet-Hein, PE van Raalte, Paul E van der Vet, and NJ Mars. “Scalability of the performance of knowledge representation systems”. In: *Towards Very Large Knowledge Bases-Knowledge Building and Knowledge Sharing* (1995), pp. 173–183.

Speer, Robert and Catherine Havasi. “ConceptNet 5: A large semantic network for relational knowledge”. In: *The People’s Web Meets NLP*. Ed. by Iryna Gurevych and Jungi Kim. Berlin: Springer, 2013, pp. 161–176.

Steyvers, Mark and Tom Griffiths. “Probabilistic topic models”. In: *Handbook of latent semantic analysis* 427.7 (2007), pp. 424–440.

Strubell, Emma, Patrick Verga, David Belanger, and Andrew McCallum. “Fast and Accurate Entity Recognition with Iterated Dilated Convolutions”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 2670–2680.

Suchanek, Fabian M, Gjergji Kasneci, and Gerhard Weikum. “Yago: a core of semantic knowledge”. In: *Proceedings of the 16th international conference on World Wide Web*. Calgary, Canada, 2007, pp. 697–706.

Şulea, Octavia-Maria, Sergiu Nisioi, and Liviu P Dinu. “Using word embeddings to translate named entities”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. 2016, pp. 3362–3366.

Täckström, Oscar, Ryan McDonald, and Jakob Uszkoreit. “Cross-lingual word clusters for direct transfer of linguistic structure”. In: *Proceedings*

of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies. Association for Computational Linguistics. 2012, pp. 477–487.

Taras, Vas, Piers Steel, and Bradley L Kirkman. "Does country equate with culture? Beyond geography in the search for cultural boundaries". In: *Management International Review* 56.4 (2016), pp. 455–487.

Tekir, Selma, Florian Mansmann, and Daniel Keim. "Geodesic distances for web document clustering". In: *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE. 2011, pp. 15–21.

Thejas, V, Abhijeet Gupta, and Sebastian Padó. "Text-Based Joint Prediction of Numeric and Categorical Attributes of Entities in Knowledge Bases". In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. 2019, pp. 1194–1202.

Tjong Kim Sang, Erik F and Fien De Meulder. "Introduction to the CoNLL-2003 shared task: language-independent named entity recognition". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. 2003, pp. 142–147.

Toutanova, Kristina, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. "Representing text for joint embedding of text and knowledge bases". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal, 2015, pp. 1499–1509.

Trouillon, Théo, Christopher R Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. "Knowledge graph completion via complex tensor factorization". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 4735–4772.

Trouillon, Théo and Maximilian Nickel. "Complex and holographic embeddings of knowledge graphs: a comparison". In: *arXiv preprint arXiv:1707.01475* (2017).

Turney, Peter D. “Similarity of semantic relations”. In: *Computational Linguistics* 32.3 (2006), pp. 379–416.

Turney, Peter D and Patrick Pantel. “From frequency to meaning: Vector space models of semantics”. In: *Journal of artificial intelligence research* 37 (2010), pp. 141–188.

Tversky, Amos and Itamar Gati. “Studies of similarity”. In: *Cognition and categorization. Hillsdale, Erlbaum* (1978).

Upadhyay, Shyam, Yogarshi Vyas, Marine Carpuat, and Dan Roth. “Robust Cross-lingual Hypernymy Detection using Dependency Context”. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. New Orleans, LA, 2018, pp. 607–618.

Van Der Maaten, Laurens, Eric Postma, and Jaap Van den Herik. “Dimensionality reduction: a comparative”. In: *J Mach Learn Res* 10.66-71 (2009), p. 13.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

Voorhees, Ellen M. “Overview of the TREC-9 question answering track”. In: *In Proceedings of the Ninth Text REtrieval Conference (TREC-9*. Citeseer. 2001.

Vrandečić, Denny and Markus Krötzsch. “Wikidata: a free collaborative knowledgebase”. In: *Communications of the ACM* 57.10 (2014), pp. 78–85.

Vulić, Ivan, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. “Hyperlex: A large-scale evaluation of graded lexical entailment”. In: *Computational Linguistics* 43.4 (2017), pp. 781–835.

Wager, Stefan, Sida Wang, and Percy S Liang. “Dropout training as adaptive regularization”. In: *Advances in neural information processing systems*. 2013, pp. 351–359.

Wang, Zhen, Jianwen Zhang, Jianlin Feng, and Zheng Chen. "Knowledge graph embedding by translating on hyperplanes." In: Citeseer. 2014.

Weeds, Julie, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. "Learning to distinguish hypernyms and co-hyponyms". In: *Proceedings of COLING*. Dublin, Ireland, 2014, pp. 2249–2259.

Weischedel, Ralph and Ada Brunstein. "BBN pronoun coreference and entity type corpus". In: *Linguistic Data Consortium, Philadelphia* 112 (2005).

Weischedel, Ralph, Sameer Pradhan, Lance Ramshaw, et al. "OntoNotes Release 4.0". In: ().

Weisstein, Eric W. "Bernoulli distribution". In: *https://mathworld. wolfram. com/* (2002).

West, Robert, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. "Knowledge base completion via search-based question answering". In: *Proceedings of WWW*. Seoul, Korea, 2014, pp. 515–526.

Westera, Matthijs and Gemma Boleda. "Don't Blame Distributional Semantics if it can't do Entailment". In: *Proceedings of the 13th International Conference on Computational Semantics-Long Papers*. 2019, pp. 120–133.

Weston, Jason, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. "Connecting language and knowledge bases with embedding models for relation extraction". In: *Proceedings of EMNLP*. Seattle, WA, 2013, pp. 1366–1371.

Wilcoxon, Frank. "Individual comparisons by ranking methods". In: *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

Wittgenstein, Ludwig. *Philosophical investigations*. John Wiley & Sons, 2009.

Wiwie, Christian, Jan Baumbach, and Richard Röttger. "Comparing the performance of biomedical clustering methods". In: *Nature methods* 12.11 (2015), p. 1033.

Woeginger, Gerhard J. “Exact algorithms for NP-hard problems: A survey”. In: *Combinatorial optimization—eureka, you shrink!* Springer, 2003, pp. 185–207.

Wu, Ting-Fan, Chih-Jen Lin, and Ruby Weng. “Probability estimates for multi-class classification by pairwise coupling”. In: *Advances in Neural Information Processing Systems* 16 (2003).

Xie, Ruobing, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. “Representation Learning of Knowledge Graphs with Entity Descriptions.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Phoenix, AZ, 2016, pp. 2659–2665.

Xu, Rui and Donald C Wunsch. “Survey of clustering algorithms”. In: (2005).

Yaghoobzadeh, Yadollah, Heike Adel, and Hinrich Schütze. “Corpus-level fine-grained entity typing”. In: *Journal of Artificial Intelligence Research* 61 (2018), pp. 835–862.

Yao, Liang, Chengsheng Mao, and Yuan Luo. “KG-BERT: Bert for knowledge graph completion”. In: *Free radical biology & medicine.* (2019).

Yarowsky, David. “Word-sense disambiguation using statistical models of Roget’s categories trained on large corpora”. In: *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics. 1992, pp. 454–460.

Yates, Alexander, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. “Textrunner: open information extraction on the web”. In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*. 2007, pp. 25–26.

Yih, Scott Wen-tau, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. “Semantic parsing via staged query graph generation: Question answering with knowledge base”. In: (2015).

Yih, Wen-tau, Geoffrey Zweig, and John C Platt. “Polarity inducing latent semantic analysis”. In: *Proceedings of the 2012 Joint Conference on*

Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2012, pp. 1212–1222.

Yogatama, Dani, Dan Gillick, and Nevena Lazic. "Embedding methods for fine grained entity type classification". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 2015, pp. 291–296.

Yosef, Mohamed Amir, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. "Hyena: Hierarchical type classification for entity names". In: *Proceedings of COLING 2012: Posters*. 2012, pp. 1361–1370.

Zadeh, Reza Bosagh and Shai Ben-David. "A uniqueness theorem for clustering". In: *arXiv preprint arXiv:1205.2600* (2012).

Zeiler, Matthew D. "Adadelta: An adaptive learning rate method". In: *CoRR, abs/1212.5701*. 2012.

Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification". In: *Advances in neural information processing systems*. 2015, pp. 649–657.

Zhitomirsky-Geffet, Maayan and Ido Dagan. "Bootstrapping distributional feature vector quality". In: *Computational linguistics* 35.3 (2009), pp. 435–461.

Zipf, George Kingsley. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, 2016.

Zirn, Cäcilia, Vivi Nastase, and Michael Strube. "Distinguishing between instances and classes in the wikipedia taxonomy". In: *European Semantic Web Conference*. Springer. 2008, pp. 376–387.