Institute for Visualization and Interactive Systems
Intelligent Systems Group
University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Institute for Natural Language Processing
University of Stuttgart
Azenbergstr. 12
D—70174 Stuttgart

Technical Report Nr. 2010/09

Methods for Coreference Visualization and Annotation

Andre Burkovski Gunther Heidemann
Hamidreza Kobdani Hinrich Schütze

CR-Classification: H.3.3, H.5.2, I.2.7, I.5.1, I.5.4

Contents

List of Figures

List of Tables

1 Introduction

One essential part in Natural Language Processing is text understanding and coreference resolution plays a major role in this challenge. In short, coreference resolution is the task of identifying entities to which the noun phrases in a text refer. This task is beneficial in NLP applications, like information retrieval, machine translation, text summarization and question answering. Coreference resolution is usually resolved with machine learning methods which require training data to learn internal parameters. English news articles or similar well-edited documents are typically utilized for the training of a coreference resolution system.

There has been a lot of work on coreference resolution using rule-based systems or machine learning systems. But since the task is a difficult one, there are still a lot of unresolved problems. One of these issues is the limited amount of available training data. One needs full understanding of the text and a basic level of linguistic knowledge to annotate a text with coreference information. This is more difficult than the annotation with other linguistic information. One goal of the software presented in this work, is to facilitate the annotation of text with coreference information.

The traditional approach for annotating documents, e.g. using text based visualizations, requires a lot of time and effort. Alternative visualization approaches can not only support the user to annotate documents, but also give insight into the coreference feature space.

Our approach combines unsupervised machine learning methods with visualization and interaction techniques to support the text annotation task and to explore the feature space. In this work, we present a visualization method based on Self Organizing Maps (SOMs). One of our goals is to enable a researcher to explore the feature space used for machine learning. This can help the feature engineer to see areas in the data where coreferent data is not clearly separable from other data. The user is able to utilize the information for the design of new features which can better solve a specific problem.

A SOM is a type of artificial neural network which projects high-dimensional input data on a low dimensional map. Thus, it is suitable as a basis for visualizations of high dimensional coreference feature space. We discuss the benefit of using SOM based visualizations for three applications regarding coreference resolution. The first application is the presentation of high dimensional coreference data and their features in a low dimensional space. This allows a better understanding of the data distribution in the feature space. The second application is the design of new features based on knowledge gathered through SOM based data exploration. The third application is an annotation task, where a user can annotate data with coreference information. This application shows that SOM based visualizations are capable of reducing the time and effort needed for annotating large documents.

1.1 Related work

Research in the area of coreference resolution mainly focused on machine learning methods for solving the task. Elango *et al.* [3] presented a survey on coreference resolution and Clark *et al.* [7] give a good overview on state of the art coreference problems and solutions. Ng *et al.* [13] presents current supervised learning and unsupervised learning models [14] for coreference resolution.

In contrast to these approaches, our focus lies on visualization methods for coreference information which support users in dealing with problems related to coreference resolution.

Some text-based approaches were developed to create visualizations of coreference information. The probably best known tool is the GATE framework [1], whose functionality is not limited to coreference annotation [2], but provides a whole linguistic engineering tool. The coreference module of GATE provides a link visualization based on plain text. The coreferences are color coded and presented to the user. However, the user has to check every generated link individually.

Another tool that was developed for coreference annotation is MMAX2 [12], which also utilizes plain text for visualization. Coreference information is presented with the help of HTML documents, where one can see dependencies between words. A framework for coreference resolution which aims at simple usage is BART [20]. The BART system is modular and the main visualization module is based on the MMAX2 system.

The Reconcile tool [17] was recently introduced, which also uses plain text for presentation of coreference information. CorefDraw [6] seems to provide a similar visualization as MMAX2 but is no longer available.

All text-based visualizations present coreference information in three ways: color coding, link identification with edges, and dependencies among words. These methods do not show the features, the feature space, or the similarity between links. Such visualizations are also limited by the size and the number of coreference lines/colors a user can distinguish. This makes it difficult to analyze large chains, inter-document coreferences or many links at once.

An alternative way to text-based visualization of coreferences was developed by Witte *et al.* [22]. Their framework presents coreferences as topic maps. Different views in the framework provide a good overview about the relationship of noun phrases in a link. The authors address the problem that the visualization of coreferences consists solely of highlighted plain text, possibly including edges for marking a coreference relation. We agree with them that textual visualization slows the user down and makes cross document annotation difficult. Nonetheless their representation only serves to visualize and navigate the result space (links), not the feature space. The noun phrases are displayed without context which makes it hard to judge whether a link is correct or not.

SOMs have been employed for visualizations in NLP before. One popular example of SOMs in NLP is WEBSOM [8], where similar documents are clustered together. Another application can be found in the lexical domain [10], where the authors used different SOMs to simulate

language acquisition of children. Outside the NLP domain, SOMs are successfully applied in various fields where unsupervised learning supports the exploration of the feature space.

Our visualization approach is similar to the method described by Heidemann *et al.* [5] where SOMs are used to cluster and label images. Images (or image clusters) are, however, much simpler to interpret by users. Therefore, the visualization based on the SOM calculation can be kept simple and is not directly comparable to NLP applications where pair-wise based coreference models are used as input.

2 Methods

Here we present a short overview of concepts surrounding our work. We introduce coreferences, explain the SOM learning method, and discuss the features we are using for the training.

2.1 Coreferences

When people talk about someone or something, they do not always use the full name to refer to this person or thing. For example the person *Bill Clinton* could be referred to by part of his name like *Clinton* or a description like *the president* or a personal pronoun like *he*. All these expressions refer to the same person and are called coreferences. The opposite of coreference is **disreference**. Two expressions that do not refer to the same discourse entity, but to two distinct entities, are disreferent. A human listener performs coreference resolution intuitively to understand what the other person is talking about.

In some cases coreference is simple to determine. It is easy to detect that the expressions *(Jordan King Hussein)*$_1$ and *(Hussein)*$_3$ in the example in Figure 2.1 are coreferent, because one is a substring of the other. On the other hand, intuitively two names that are not the same (like *Hussein* and *Clinton*) can never be coreferent. For human readers the connection of *(the president)*$_4$ with the previously mentioned entity *(U.S. President Bill Clinton)*$_2$ is obvious. Although, if expression 2 would only consist of *(Bill Clinton)*$_2$ we (as human readers) would need to know that he is (was) the president or the text would have to contain that information somewhere.

To resolve a pronoun like *(his)*$_5$ we also need the context to decide whether it refers to *Hussein* or *Clinton*.

> *The White House said on Monday* ***(Jordan King Hussein)***$_1$ *would meet* ***(U.S. President Bill Clinton)***$_2$ *in Washington on April 1 and denied that the Middle East peace process was unraveling.* ***(Hussein)***$_3$ *had been scheduled to meet* ***(the president)***$_4$ *on March 18, but* ***(his)***$_5$ *visit was postponed after a Jordanian soldier shot dead seven Israeli girls near the Israel-Jordan border on March 13 and after* ***(Clinton)***$_6$ *had knee surgery on March 14.*

Figure 2.1: Example for Coreference (from the ARE Corpus)

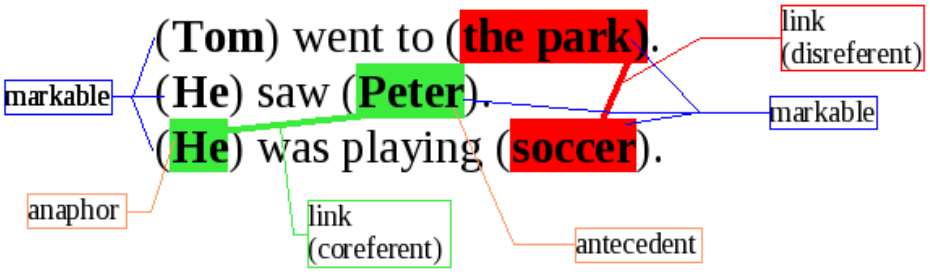


Figure 2.2: Basic Terminology

2.1.1 Markables and Links

An expression that might be coreferent to another expression is called a **markable**. All markables we are going to consider for coreference resolution are noun phrases. Simplifying slightly, we can say that a **noun phrase** is a group of words in a sentence that can be replaced by a (pro-)noun. For example *the house* or *my small yellow old house that my father built when I was a kid* can be replaced by the pronoun *it*.

Often a markable A is coreferent with another markable B and B is coreferent with yet another markable C. Such sets of markables which refer to the same entity are called **coreference chains**.

A pair of markables that can be co- or disreferent is called a **link**. This link has a number of associated link features. A link can have a **label** that contains information about co-/disreference of that link. The text in Figure 2.2 contains two labeled links (one coreferent and one disreferent).

The first markable in a link (in order of appearance in the text) is called **antecedent**, the second one is called **anaphora**. Examples for both are given in Figure 2.2. A link is created by linking a markable (the anaphora) with another markable which occurs earlier in the text.

2.2 Self Organizing Maps

A Self Organizing Map (SOM)[9] is an unsupervised machine learning method. Since SOMs project high dimensional input data to a low dimensional output space (map) they are popular as a basis for different visualizations.

A SOM is a neural network where neurons (or nodes) are connected to each other by a low dimensional topology. Every node $n_i \in N$ of the SOM has a definite location $r_i \in \mathbb{R}^{d_{topol}}$ in the topology of dimension d_{topol}. The most common topology is a platonic tessellation – a two dimensional grid of equilateral triangles, squares or hexagons. Each node has a

corresponding weight vector $w_i \in \mathbb{R}^{d_{in}}$ of the same dimension as the input data d_{in}. In the training phase, the weight vectors adapt to the input data. The learning rule is a modification of the Winner-Takes-All rule with additional adaption factor.

For every input vector $\vec{x} \in \mathbb{R}^{d_{in}}$, the distance $d(\vec{w}_i, \vec{x})$ for all nodes n_i is calculated. The winner node n_k, where $k = argmin_i(||\vec{x} - \vec{w}_i||)$, with the minimum distance to a given input vector $\vec{x}$, is called *best matching unit* (BMU).

Weight vectors w_i of all nodes are updated according to the learning rule

$$\Delta\vec{w}_i = h_{ik}\alpha(\vec{x} - \vec{w}_i)$$

where k is the BMU, α is a time decreasing learning coefficient and h_{ij} is the neighborhood function.

Any neighborhood function h_{ik} can be chosen, however, usually a Gaussian is used. The learning process of the SOM preserves the topological structure of the input data in the resulting map, i.e. feature vectors which are similar in the input space will be close together in the output space. Graphically speaking, the weight vectors change their places in feature space to get closer to the data points and take their neighbors with them.

It is worth noting that for training we use the software package for SOMs implemented by Kohonen's group: the SOM-Toolbox for Matlab v2.0 [21].

2.3 Model

In this section, we introduce the data model and features we use to train the SOM. The internal linguistic component provides the necessary data and data structures. The component uses a relational data base for data organization. It provides essential processes for visualization, like preprocessing of textual data, markable detection, link generation, and feature extraction.

Before the feature calculation, a filtering process is applied to reduce the number of links presented to the user. The idea is that links, which are certainly disreferent, are removed, e.g. links where a reflexive pronoun is linked with a markable from a different sentence.

The filter can also be used to limit links to a certain category. It might also be useful to limit the distance of two markables which can form a link or to consider only coreference inside one document to reduce the amount of data which needs to be annotated.

In our data model we use markable attributes that can be nominal, like part of a speech tags of content words, or numerical, like sentence number to calculate the link features. We introduce the features in the next section.

Word overlap	The number of words the two markables have in common.
Head match	1 if the head of markables matches, else 0.
Prenominal modifier overlap	Number of prenominal modifiers the two markables have in common.
Markable span	1 if one markable spans the other, else 0.
WordNet distance	Jaccard coefficient of WordNet hypernym sets for both markables.
Apposition	1 if the two markables are in an appositive construction, 0 if the markables fulfill the position requirements, but not the semantic, and -1 else.
Number	The number agreement of the markables.
Semantic Class	The semantic class agreement of the markables. The class is retrieved during the preprocessing.
Pronoun singular 1	1 if the first markable is a singular pronoun, 0 else.
Pronoun singular 2	1 if the second markable is a singular pronoun, 0 else.
Pronoun plural 1	1 if the first markable is a plural pronoun, 0 else.
Pronoun plural 2	1 if the second markable is a plural pronoun, 0 else.
Prenominal modifier 1	The number of prenominal modifiers of the first markable
Prenominal modifier 2	The number of prenominal modifiers of the second markable
Word frequency 1	The number of occurrence of the first markable in the text
Word frequency 2	The number of occurrence of the second markable in the text

Table 2.1: The table gives an overview of link features we extract for each link. The features are used in the training of the SOM.

2.4 Features

We implemented features inspired by Ng *et al.* [13]. The set of features we use for visualizations in this report is given in Table 2.1.

The *head match* feature uses the head word of a markable. The head is calculated as the last word in the first noun cluster in the noun phrase. Alternatively, it is the last word of the noun phrase if it does not contain a noun cluster [16].

Prenominal modifiers are all words that occur before the head that are adjectives, gerunds, past participles or other nouns [11].

As *Wordnet distance* we use the normalized symmetric difference distance [23] of hypernym sets for both markables, also known as the Jaccard coefficient. All hypernyms of the head words of both markables are retrieved for the calculation. The intersection of all hypernyms for both words is divided by the number of elements in the union of both sets. The closer the result is to 1, the more hypernyms are the same. It is 0 if there are no common hypernyms. In WordNet we use all word senses without disambiguation.

3 System Description

3.1 System Overview

We designed the system with modularity in mind. We expect the development of various visualization modules to extend the system.

The basic user interface with a simple U-Matrix visualization is shown in Figure 3.1. The user interface consists of a visualization view, as well as views for SOM parameters, SOM calculation, and additional feature information. The additional feature information shows the feature vector number assigned to the selected node, as well as the codebook vector, coreference, and disreference information, if the user provided a gold standard text.

The SOM provides multiple visualizations from which the user is able to interpret the data distribution. The lack of direct user interaction is a major drawback of the Matlab tools. We developed an interactive tool which uses the U-matrix and component planes for the visualizations and presents coreference data accordingly. Our tool aims at simple navigation and interaction with the SOM.

For annotation purposes we implemented a simple text-based coreference visualization and a graph-based link visualization, which will be described in detail in Sections 3.3. These modules assist the user whenever he or she wants to inspect the actual links and their words and sentences.

3.1.1 SOM Training Module

The SOM training module allows the user to adapt training parameters for the SOM in the UI. They are basically a subset of available parameters of the Matlab SOM-toolbox. The UI element is shown in Figure 3.2.

The calculation parameters provide settings for internal processing.

- **Calculation ID**: The ID refers to the calculation ID in the database calculation model. It allows a storage of calculation history. By default, the application determines the parameter automatically.
- **Normalization**: Normalization parameter for Matlab SOM-toolbox preprocessing.

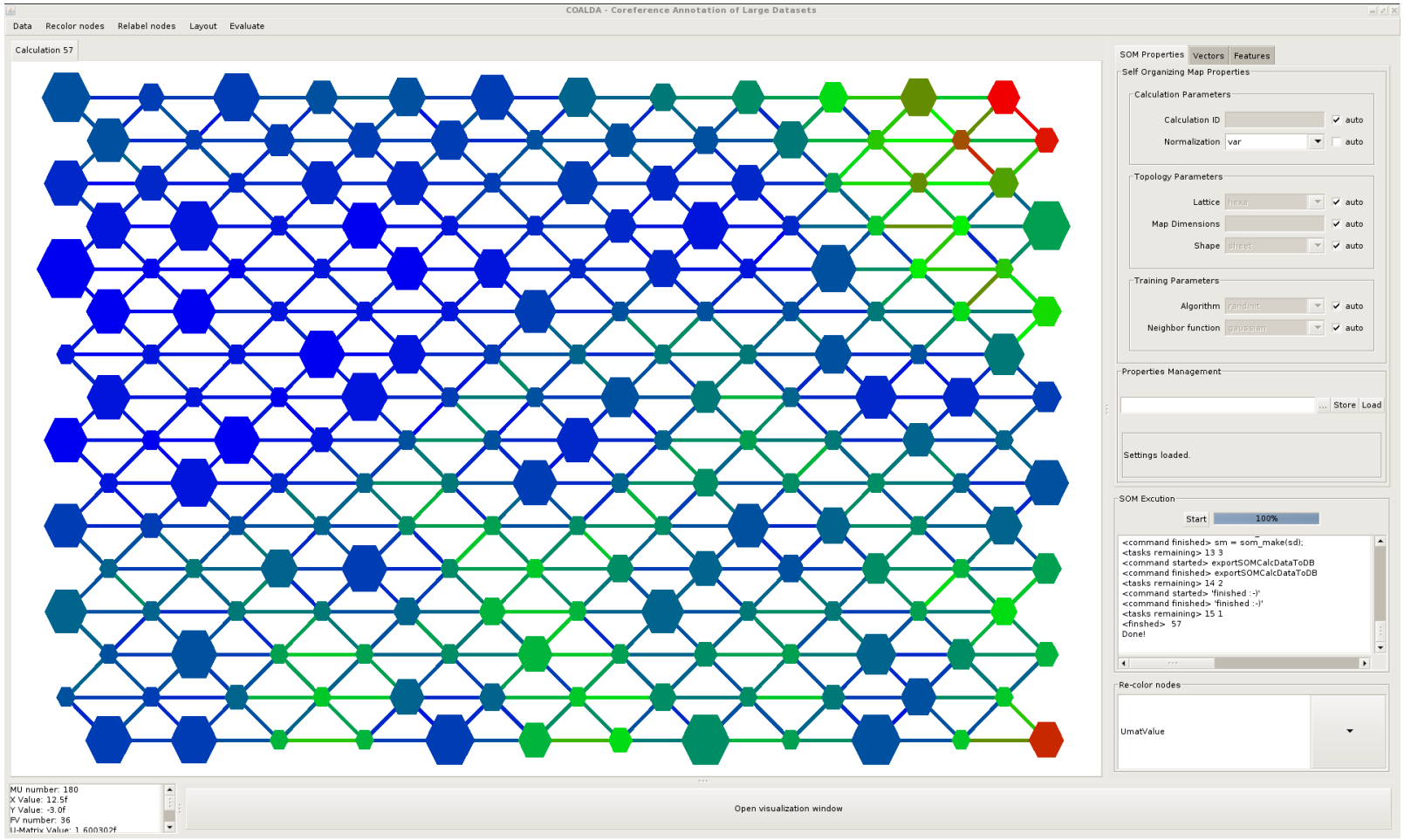


Figure 3.1: The GUI of our software system for annotation and visualization of coreference data. The central element is the U-Matrix graph. The user is able to select nodes or edges of this graph and inspect the content of nodes.

The topology parameters provide settings for the map topology.

- **Lattice**: The lattice parameter is responsible for the connectivity of the SOM nodes. The *rect* value will create a rectangle grid (four node neighbors). The *hexa* value will create a hexagonal grid (six node neighbors). By default, the parameter is set to the *hexa* value.
- **Map Dimensions**: The parameter determines the number of nodes in a SOM. It will create an $n \times m$ grid of nodes. By default, Matlab SOM-toolbox automatically chooses a value.
- **Shape**: The parameter determines the shape of the SOM. The *sheet* value creates a simple 2-D SOM shape. The *cylinder* parameter connects the bottom and top SOM nodes together. The *toroid* parameter additionally connects the left- and right-most nodes together.

The training parameters provide settings for the SOM training.

- **Algorithm**: The algorithm parameter provides a choice for the training function of the SOM. The *sequential* algorithm chooses the data at random for the training. The *batch* algorithm considers all data at once. By default, Matlab SOM-toolbox uses the *batch* algorithm. In contrary to the *sequential* algorithm, the *batch* algorithm returns always the same result for the same data (given the same parameters).
- **Neighborhood function**: The parameter determines the kernel function for the neighborhood function.

Additionally, the module contains a *Property Management* component. This component stores and loads the training parameters. In this way the user is able to save a training configuration for the SOM for later usage.

The bottom component shows the log screen for the Matlab training session. From that component the user is able to recognize errors in the execution, as well as the calculation id, that Matlab used to store the SOM training results.

3.1.2 SOM Zoom

Our tool also contains a so called SOM-zoom. This is similar to the technique used in hierarchical SOMs [15]. In hierarchical SOMs, a small map is trained and data is assigned to the nodes. Then, for every node separately, the SOM is trained again on the data. The training is finished when a desired hierarchy level is reached or another stop criterion is satisfied. Our method differs in such a way that new SOMs are trained on nodes selected by the user. For every subset of map nodes the user can train a new SOM. The SOM is then trained only with the selected data of the selected nodes.

Another feature is the selection of dimensions for the training. The user is able to select a subset of dimensions for the training. The selected dimensions are extracted from the data and create a new data set for the training. In such a way the user is able to experiment with feature dimensions and identify the features best-suited for SOM training.

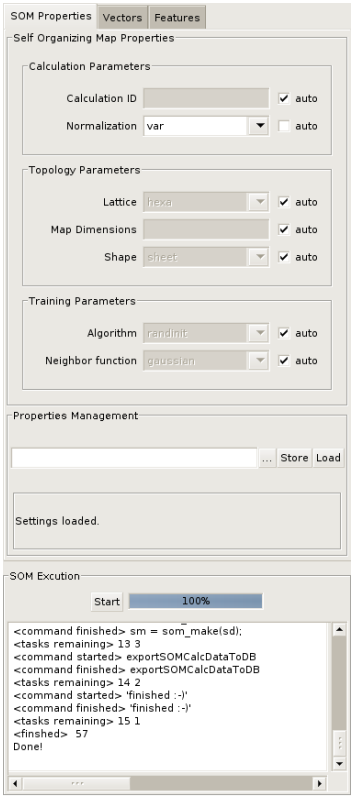


Figure 3.2: The SOM training module. In this UI element the user can set training parameters for the SOM calculation.

3.2 SOM-based Visualization Modules

The central visualization module in the application is the SOM visualization. Initially the view shows a U-Matrix for the trained SOM the calculation id (in a tab) which Matlab used to store the results in the database.

The user is able to interact with the SOM in several ways. First, the user is able to switch the visualizations and show component planes of the SOM. Second, we provide the function to show labels about the data in the nodes. For a gold standard text we show the amount of coreferent, disreferent and unknown labels. The data with unknown labels is due to the link generation process. In this process the algorithms for the link creation may create links that are not available in gold standard corpus. Third, the user may assign labels to the nodes. The label is then applied to the data in such nodes. This method allows a fast annotation of all data in the SOM nodes.

3.2.1 U-Matrix

Basic principle
The most common visualization method for the SOM is the U-matrix [19] (unified distance matrix). The topology defines the position of nodes and a neighborhood relation between nodes, represented by edges, thus providing a $n \times m$ grid of numbered nodes $k = 1..|N|$, where N is the set of map nodes. This grid is used to define the U-matrix $U \in \mathbb{R}^{(n*2-1)\times(m*2-1)}$ whose components represent nodes and edges. For example, given a 2×3 map grid with nodes $n_1, n_2, n_3, n_4, n_5, n_6$ and corresponding weight vectors w_i the U-matrix would look like

$$\begin{pmatrix} u_1 & u_{12} & u_2 & u_{23} & u_3 \\ u_{14} & u_{15} & u_{25} & u_{26} & u_{36} \\ u_4 & u_{45} & u_5 & u_{56} & u_6 \end{pmatrix}$$

where the U-matrix value for an edge $u_{ij} = ||w_i - w_j||$ is the distance in feature space of the two nodes of that edge. The U-matrix value of a node u_i can be set arbitrarily, but is normally set to the mean U-matrix value of all edges from this node.

U-Matrix as a Graph
The U-matrix is intended for an intuitive representation of the distances between nodes. The usual approach to visualize the U-matrix is to display cells for both nodes and edges. Our visualization of the U-matrix was adapted to treat the U-matrix as a graph. Instead of using cells, the SOM grid itself is used for the U-matrix visualization, as can be seen in Figure 3.3.

The nodes represent the nodes of the SOM topology. In training, the algorithm assigns BMU to the data and such BMUs are such nodes. The edges represent the connections between the nodes.

A common color-scheme is used with red for high values and blue for low values. The values in between are colored green.

3.2.2 Component Planes

Instead of coloring nodes according to the U-matrix values, it is also possible to color them based on the nodes' weight values in a single feature vector component. Section 2.4 give an overview of the features used in our coreference visualization. On of these features can be considered as one such component.

Component planes are useful for visualizing the influence of one feature on the cluster formation. Figures 3.4-3.6 shows all component planes for the features described in Section 2.4). The component planes allow a fast overview of which features dominate which region of the SOM.

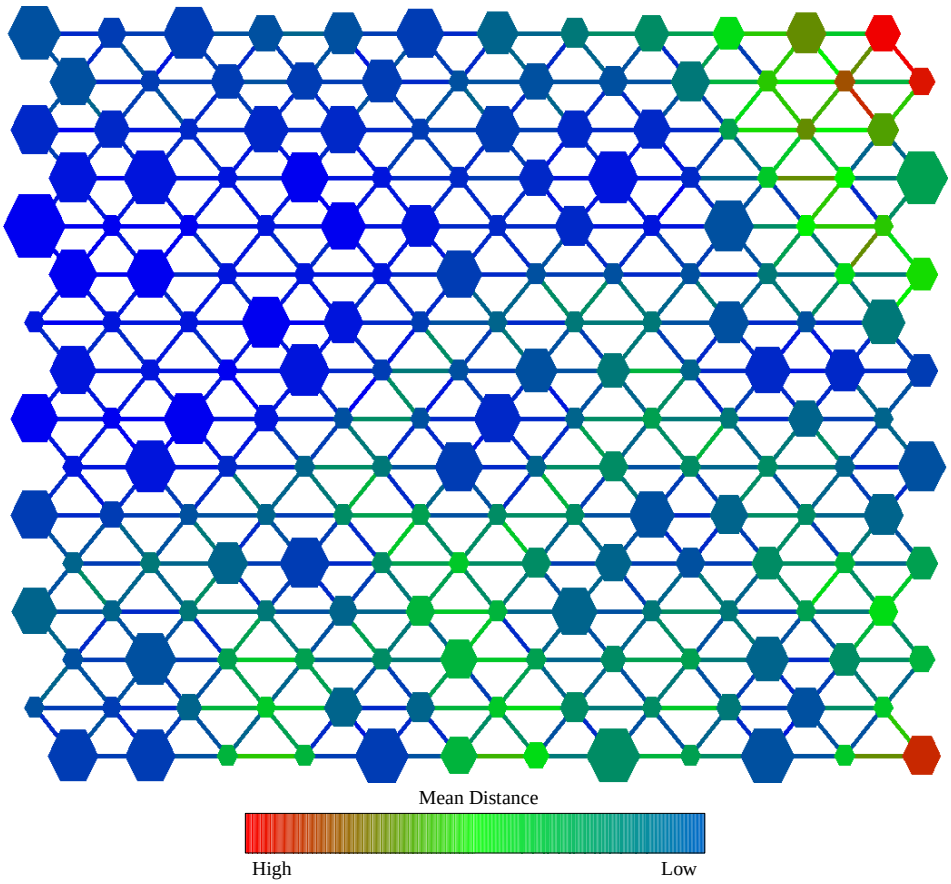
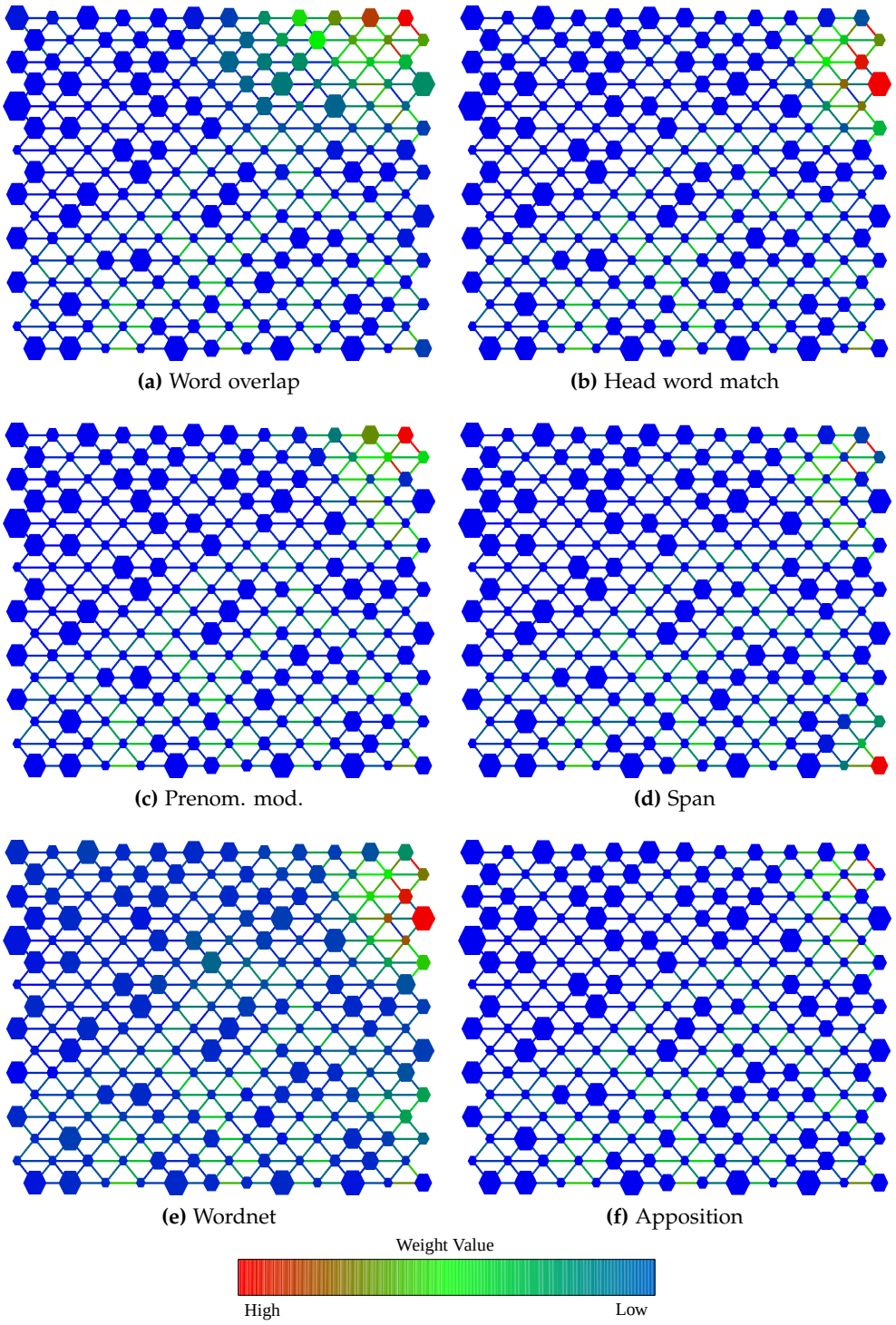


Figure 3.3: Graph-based U-matrix visualization combined with a hit histogram representation for nodes. Distances between nodes are color coded and allow visual cluster identification. The size of nodes indicates the number of feature vectors assigned to this node in the training process, i.e. the number of feature vectors for which this node was the BMU.

3.2.3 BMU Connections

We implemented a method for cluster visualization proposed by Tasdemir *et al.* [18]. Basically, the visualization displays BMU dependencies between nodes. Nodes are connected based on the the data they contain. For each data point (in our case the feature vector of the link) a second, third, and fourth BMU is calculated. An edge connects the first BMU to the other BMUs, respectively.

In Figure 3.7 only the connection between the first and the second BMU are drawn. This visualization allows the identification of homogeneous clusters, like in the upper middle and heterogeneous areas, like the lower right corner.

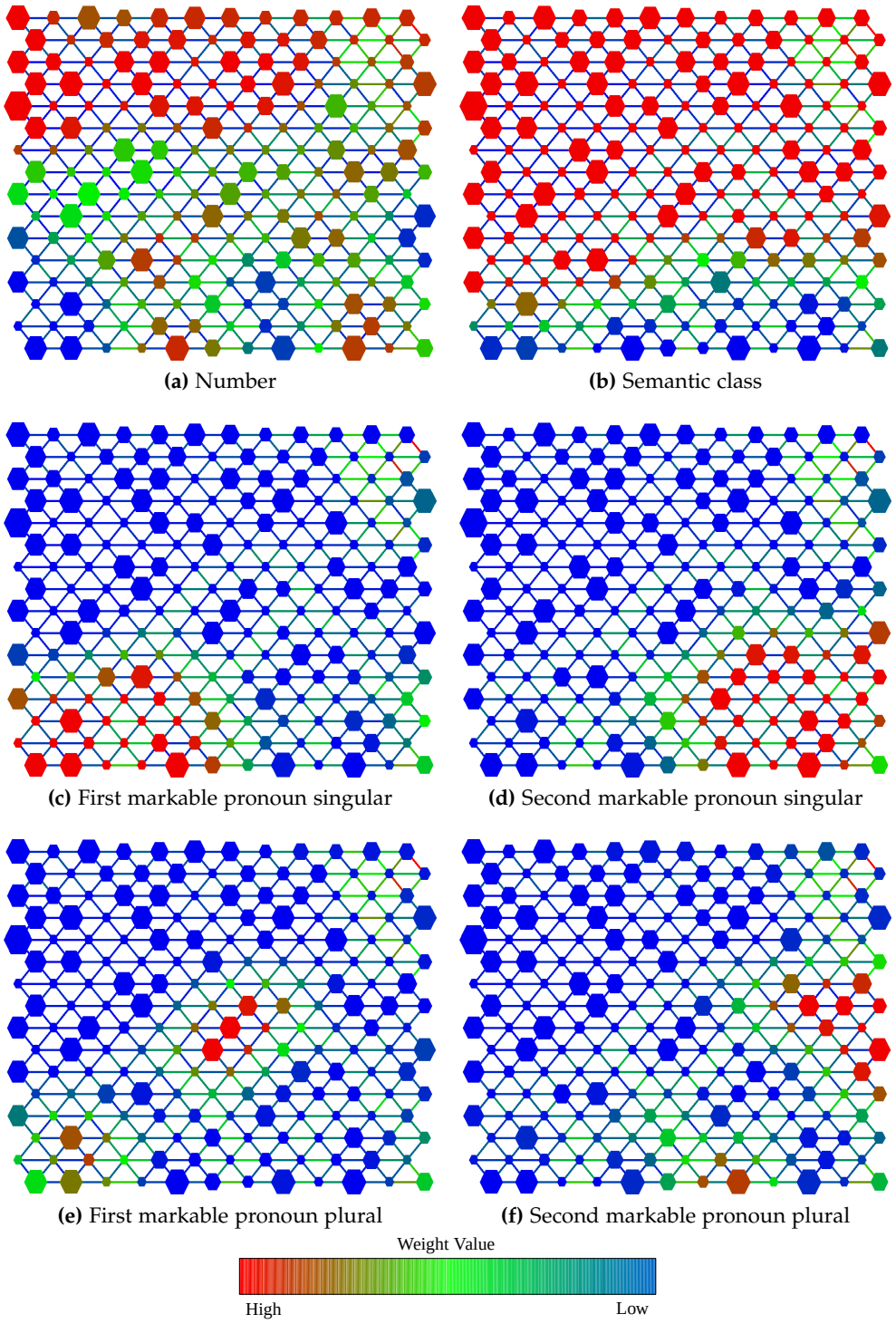


(a) Word overlap **(b)** Head word match

(c) Prenom. mod. **(d)** Span

(e) Wordnet **(f)** Apposition

Figure 3.4: Component Planes of the features 1–6: The figures 3.4a and 3.4b show the component planes for strong coreference features. The user will find many coreferent links in the upper right corner of the map, because these features have a high influence in that region. On the other hand, the component plane for the Apposition feature (Figure 3.4f) indicates that none of the links are in an appositive construction because of its low influence in the whole map.

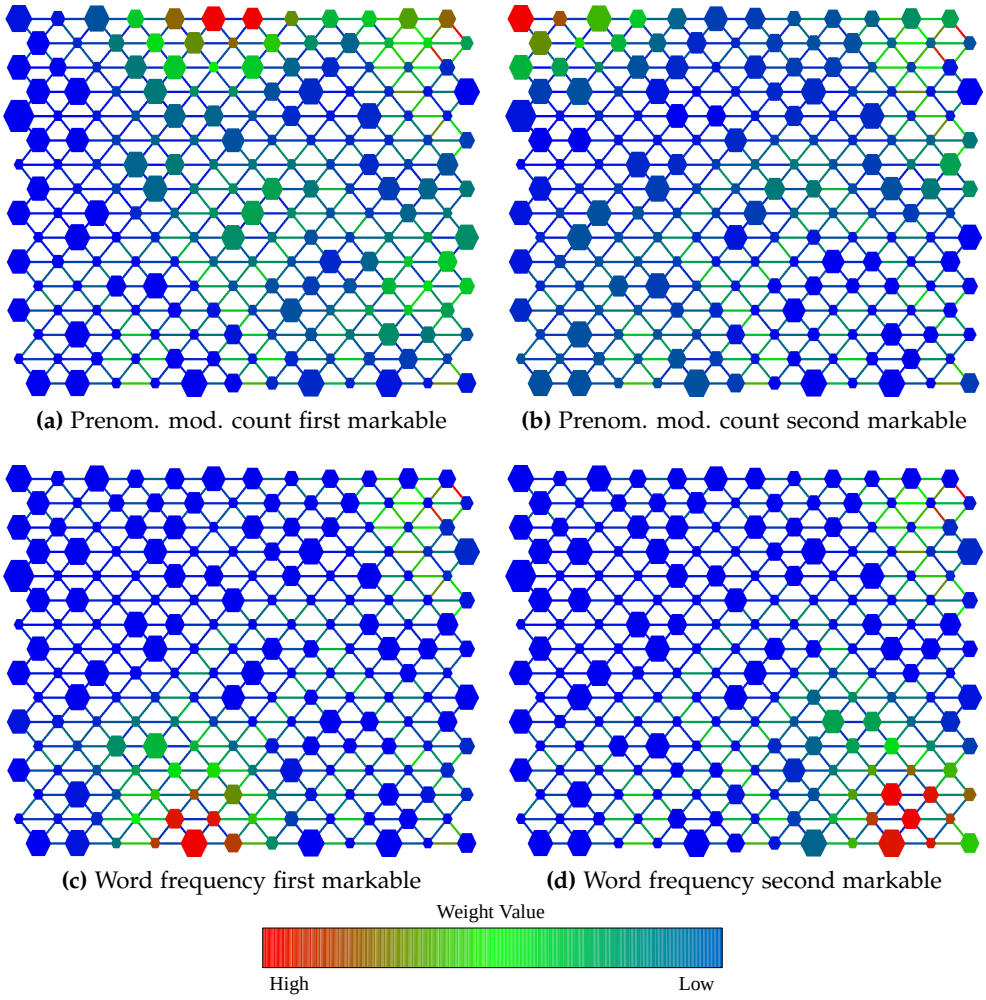


(a) Number **(b)** Semantic class

(c) First markable pronoun singular **(d)** Second markable pronoun singular

(e) First markable pronoun plural **(f)** Second markable pronoun plural

Figure 3.5: Component Planes of the features 7–12: Figures 3.5c and 3.5f show the component planes for pronouns. In the regions where the features have a high influence one will find links, where at least one markable is a pronoun (singlar or plural, respectively). In Figure 3.5a red regions indicate where markables agree in number.



(a) Prenom. mod. count first markable

(b) Prenom. mod. count second markable

(c) Word frequency first markable

(d) Word frequency second markable

Figure 3.6: Component Planes of the features 13–16: Figures 3.6a and 3.6b show the component planes for markables with pronominal modifiers. In the red regions the user is likely to find markables with many prenominal modifiers. The component planes in figures 3.6c and 3.6d show regions where the word frequency of the markable is high in a document.

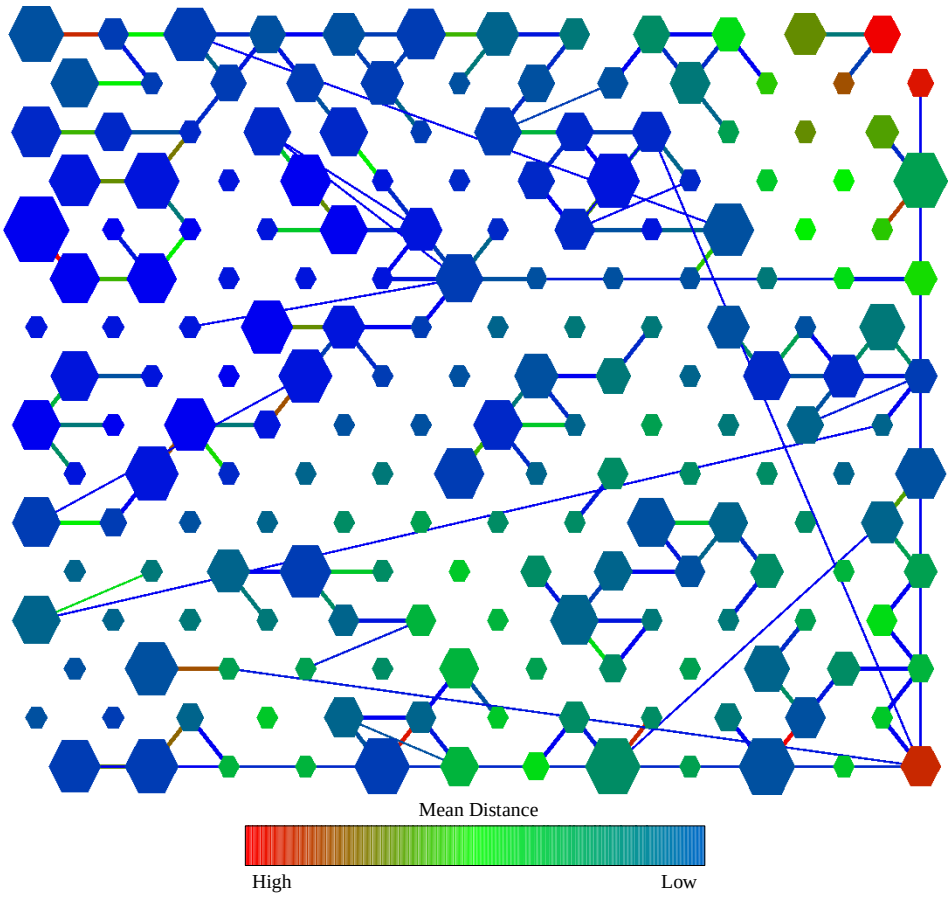


Figure 3.7: BMU connectivity visualization. Here only, the second BMU is connected with the first BMU. One can easily identify homogeneous clusters like the upper middle and heterogeneous areas like the lower right corner.

3.2.4 Force-Directed Layout

Another interesting and novel visualization is a force-directed layout of the U-Matrix. In our model we consider the U-Matrix as a graph and as such we are able to apply different graph-drawing and layout algorithms. One such layout algorithm was developed by Fruchterman *et al.* [4].

This method considers the connection between nodes as springs. Since edges in our SOM visualization encode the distance between SOM nodes we can use this coefficient as property of the spring in the graph.

The result is an organic layout of the U-Matrix graph. Figure 3.8 shows an aesthetic cluster formation. Dense cluster areas are better visible than in the original color coded U-matrix visualization.

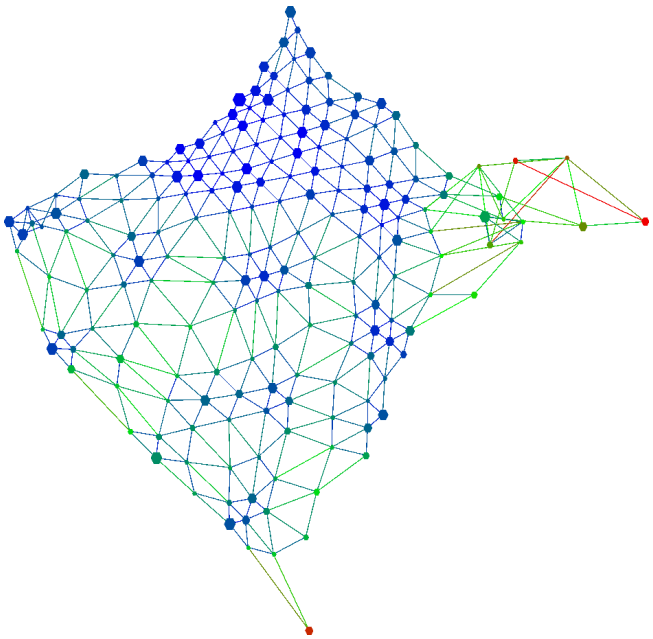


Figure 3.8: A force directed layout of the U-Matrix-Graph. Dense cluster areas, for example, at the top of the grid, are better visible than in the original color coded U-matrix visualization (Figure 3.3).

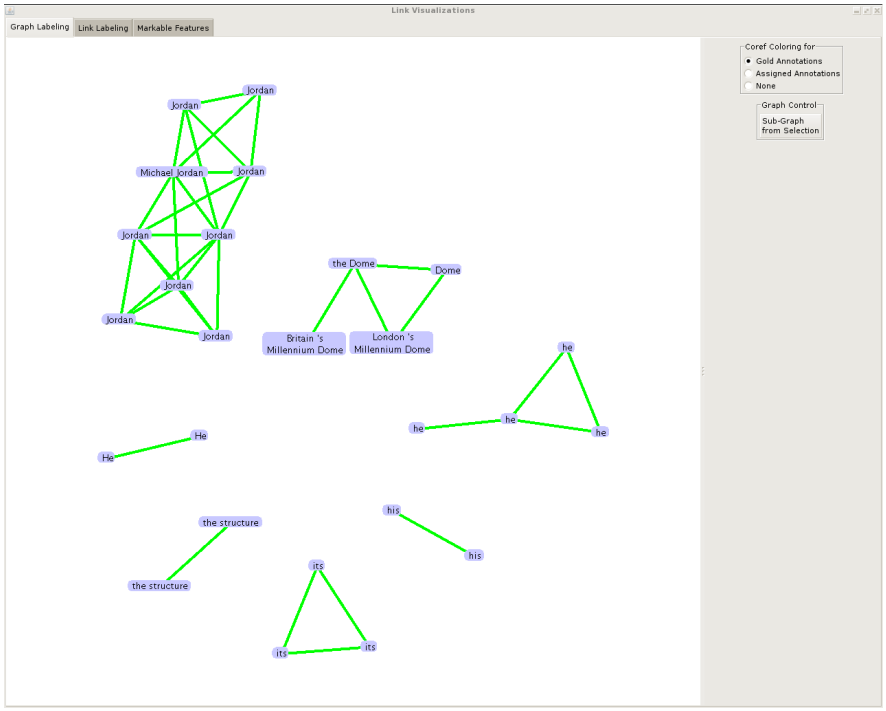
3.3 Visualizations for Annotation Support

The SOM is an abstract visualization method. In order to use the SOM for the coreference annotation task we provide several specific visualizations methods. These visualizations are displayed in a separated view and are meant to show the actual links of a SOM node selected by the user.

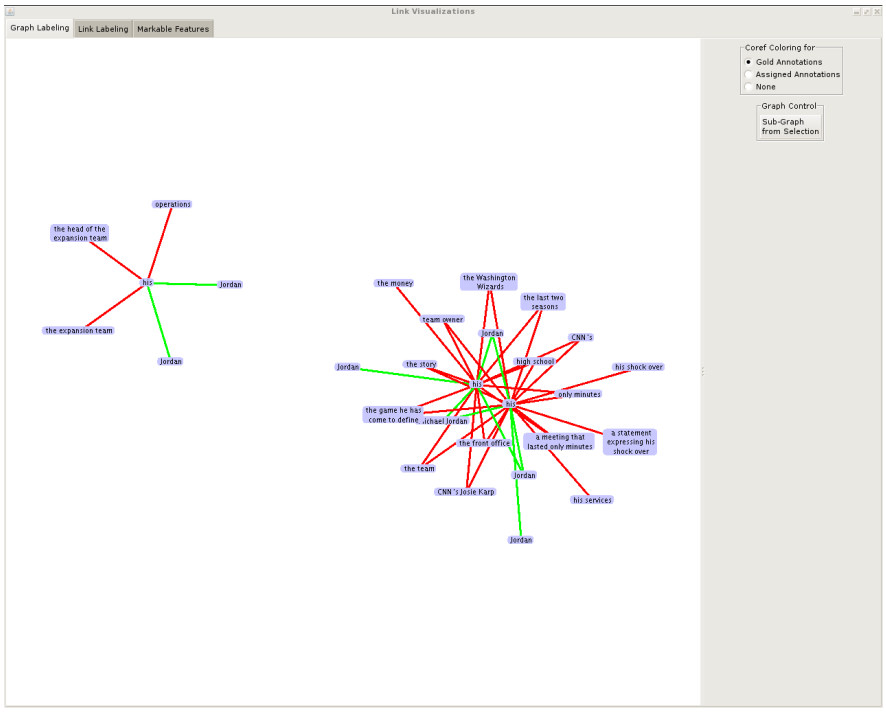
3.3.1 Link Visualization

The first visualization is a graph-based visualization of links. Some of the links of a SOM node may have transitive connections. The graph-based link visualization shows these connections (Figure 3.9). In the visualization an edge represents a link and the node represents a markable.

The user can interact with the graph in several ways. He or she can select nodes (all outgoing edges are selected automatically), edges, or subgraphs. The user may then annotate all selected links (edges) with coreference information.



(a)



(b)

Figure 3.9: Link visualization of two different SOM nodes. The contents of one node shows coreferent link relationships (Figure 3.9a). The contents of another node show highly connected and mixed links. Such nodes are difficult to annotate with a graph-based visualization.

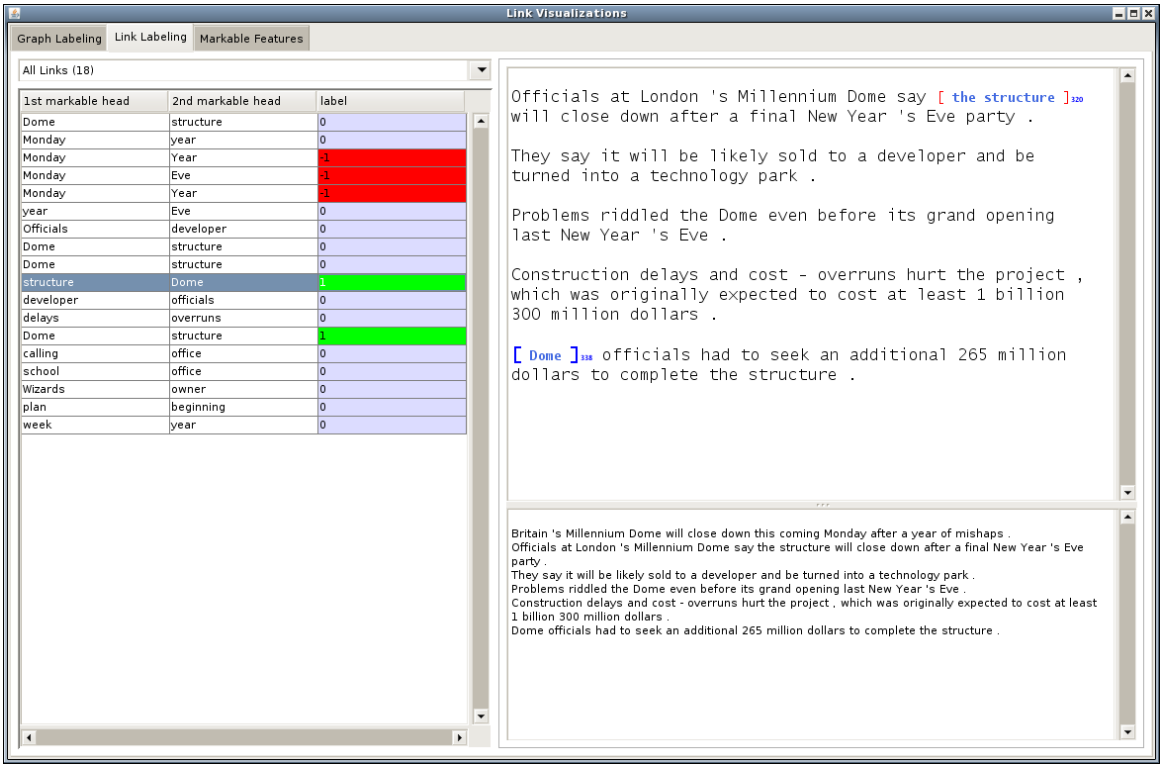


Figure 3.10: The text-based coreference annotation module. The user interacts through elements in the table. The right panel shows the text, the links are embedded in. The user is then able to derive the coreference status and annotate the link accordingly.

3.3.2 Text Visualization

The purpose of the text-based visualization is to present the text surrounding the links to provide context information. We developed two methods – one for the annotation of links and one that utilizes the gold-standard text to display links and their coreference labels. Figure 3.10 shows a table of selected links and the text they are embedded in. The user may inspect each element in the table and annotate it with coreference information.

Figure 3.11 shows the gold-standard text, which displays the links and their coreference labels. Additionally, the user is able to inspect the feature vectors of the links.

3.4 Applications

This software enables computational linguists (and alike) to gain insight into the coreference feature space and allows several methods for annotation. The SOM-based method, where the user can assign labels to nodes, in which case all links in a node are labeled accordingly. The

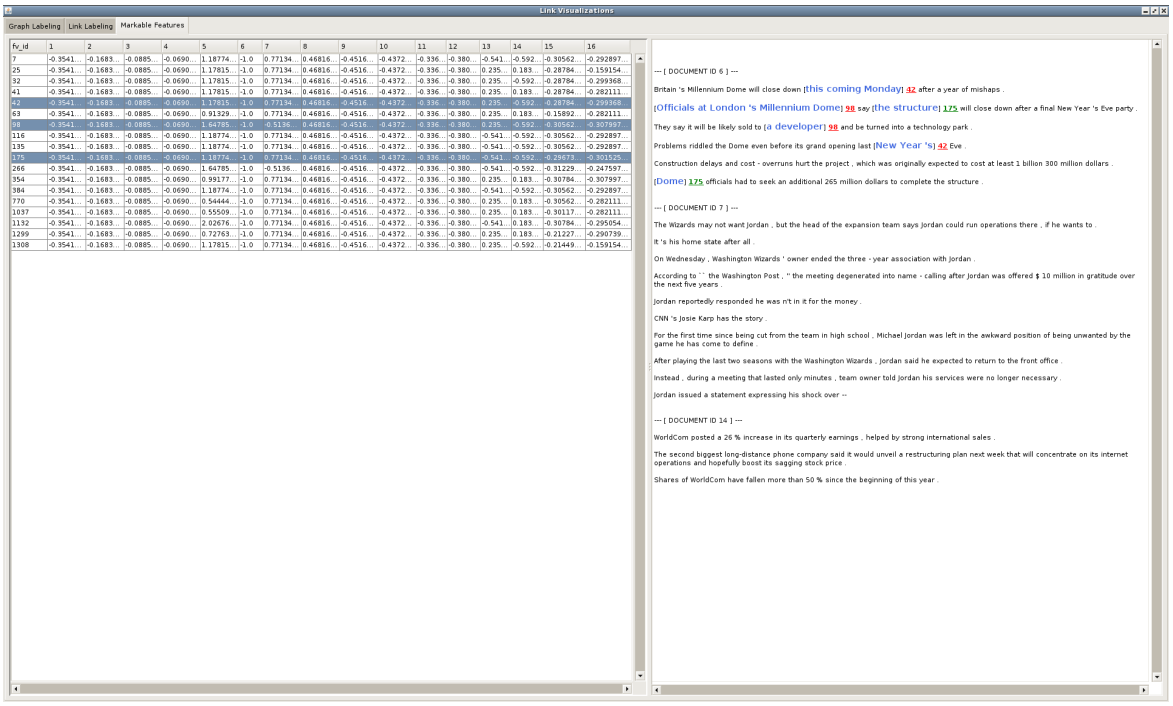


Figure 3.11: This visualization shows the feature vectors (table) of the links and their textual representation.

graph-based method, where the user is able to annotate edges of links. And the text-based method, which basically shows links in a text. The main motivation for the development of the tool comes from the wish to understand the coreference formation better and the need to annotate large data sets. We present three applications for this tool: *feature space exploration*, *feature engineering*, and *annotation*.

3.4.1 Feature space exploration

Utilizing our system, users can always investigate which links are assigned to a node by selecting that node. In addition, with annotated text, the user can see which links are coreferent or disreferent. The user can then color-code the nodes of the map with information about the proportion of dis- and coreferent data. In such a visualization, the color indicates the proportion of coreferent links to disreferent links.

Usually, data clusters are created by the influence of one or more features. Clusters are isolated by nodes with no feature vectors or edges, where the U-matrix value is high (red edges). Features which are responsible for the cluster can be identified, using the component planes.

The component plane of one feature, a simple head match, is shown in Figure 3.4b. Head match is a good indicator for coreference and the experienced user may find many coreferences in such a cluster.

Figure 3.4e shows the component plane of the WordNet distance feature. Again, high values have a red color and one can conclude, that some links in the WordNet cluster may be coreferent. There are value for which the WordNet distance is undefined. This is the case if no WordNet synsets were found for the phrase; a common value if the words are both proper names. The same deduction technique also can be used to detect disreferent feature vectors.

Feature space exploration via the component planes enables the user a fast judgment of how well the map has clustered the data.

3.4.2 Feature Engineering

The feature space exploration gives a good insight into how well the features are suited for the SOM. The user can identify clusters of nodes where the separation of the data is not clear. The user just may activate the labels for the nodes. In such case the nodes show how many coreferent and disreferent links (given the gold-standard text) they contain.

In some regions the user may find heterogeneous nodes. These nodes contain coreferent links, but also have some disreferent links as well. This indicates that new features should be developed to better separate coreferent and disreferent links. The user can inspect these nodes and view the markables and related text for the links assigned to this node. This allows the user to understand what the markables have in common and why they were assigned to the same node.

In our tool the user has the option to easily recalculate the SOM for a subset of the features. In an experiment we recalculated the SOM for only a handful of the original features. Some feature vectors that were distributed over several nodes in the original SOM were now concentrated in much fewer nodes. Thus one can experiment with a feature set and find features which influence the clustering in a positive or negative way.

It is also possible to apply the SOM-zoom for a node. This is very useful for nodes where the assigned data vectors have a high variance in their components and the weight vectors also have high values in each component. In the new resulting SOM, the map nodes have a different topology. Thus, the data is reordered accordingly and the user may recognize some clusters more easily.

The inspection of nodes with mixed links helps the user to understand what these links have in common and which new feature may separate them. The recalculation of a new SOM for mixed nodes and a subset of features used may result in a different, better clustering of the links.

3.4.3 Annotation

Using our visualizations the user is also able to annotate the links. The SOM reveals good clusters of only disreferent and coreferent links. Using the described visualizations, the user may select and annotate the data with coreference information. The user is able to identify clearly disreferent and coreferent clusters for individual component planes, with additional help through textual representation. Additionally, a strong indicator for coreference, like the head match feature, helps recognizing clusters. Thereby the user is able to annotate whole clusters of data with appropriate coreference information.

Once the user realizes the influence of component planes for features that are a strong indicator for coreference or disreference, he or she is able to annotate new data. For new unannotated data, the user can train a new SOM and apply the acquired knowledge of component planes to annotate a set of nodes with coreference information.

4 Conclusion

In this work, we presented visualization techniques for our interactive user interface for the exploration and annotation of coreference information. It uses Self Organizing Maps (SOM) to create a low-dimensional representation of high-dimensional feature data

We introduced visualization modules and described their use in several real-world applications concerning coreference resolution in NLP. First, the low dimensional presentation space of the SOM enables the user to explore the high dimensional feature space. This conveys a better understanding to the user of how the data is organized and why. Second, the SOM allows the user to judge the quality of the features and to explore nodes with mixed coreference information. The content of these nodes can help the user to design new features. The third application enables the user to annotate many links at once. It also allows the assignment of user-specified confidence values to labels.

Future Work

Since this work presents a new approach to visualization of coreference, there are still many ideas and challenges.

One possibility to enhance the visualization is to include data density information. This could be done by calculating the P-matrix value and showing this value instead of the U-matrix value of the nodes. This corresponds to visualizing the U*-matrix.

Another idea is to visualize selected features as icons. Areas where the weight values of the nodes in this feature are high can then be identified by the icon.

To give the user a starting point for labeling, some links could be labeled before displaying the visualization. This prelabeling can be done by the user, or automatically using "reliable features" such as exact string match (nearly always coreferent) or span (nearly always disreferent) and marking the links where these features have the value `true`. In the visualization, the positions of the prelabeled links are then indicated.

Due to the transitivity of the coreference relation, as soon as the user has labeled some links, many other links are known to be coreferent or disreferent. If a user labels link A-B as coreferent when link B-C is coreferent, this entails that link A-C is coreferent. This could be used to automatically label all transitive links found in the SOM.

Since the fully automatic labeling of coreference information is still far out of reach, the extension of visual analytics techniques will be a topic for future research. Coreference

resolution is a very complex topic and will require major extensions and adaptations in order to provide benefit to NLP developers.

Bibliography

[1] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics* (2002). (Cited on page 6)

[2] Dimitrov, M. Light-weight approach to coreference resolution for named entities in text. Master's thesis, University of Sofia, 2002. (Cited on page 6)

[3] Elango, P. Coreference resolution: A survey. Tech. rep., Unversity of Wisconsin Madison, 2005. (Cited on page 6)

[4] Fruchterman, T. M. J., Edward, and Reingold, E. M. Graph drawing by force-directed placement. *Software — Practice and Experience 21* (1991), 1129–1164. (Cited on page 22)

[5] Gunther Heidemann, Axel Saalbach, H. R. Semi-automatic acquisition and labelling of image data using soms. In *European Symposium on Artificial Neural Networks* (April 2003), pp. 503–508. (Cited on page 7)

[6] Harabagiu, S. M., Bunescu, R. C., and Trausan-Matu, S. Corefdraw: A tool for annotation and visualization of coreference data. In *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence* (2001). (Cited on page 6)

[7] Jonathan H. Clark, J. P. G.-B. Coreference: Current trends and future directions. Tech. rep., The Language Technologies Institute, CMU, 2008. (Cited on page 6)

[8] Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. Websom - self-organizing maps of document collections. *Neurocomputing 21* (1997), 101–117. (Cited on page 6)

[9] Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics 43*, 1 (Jan. 1982), 59–69. (Cited on page 10)

[10] Li, P., Farkas, I., and MacWhinney, B. Early lexical development in a self-organizing neural network. *Neural Networks 17*, 8-9 (2004), 1345 – 1362. New Developments in Self-Organizing Systems. (Cited on page 6)

[11] Mitchell, M. Class-based ordering of prenominal modifiers. In *ENLG '09: Proceedings of the 12th European Workshop on Natural Language Generation* (2009). (Cited on page 12)

[12] Müller, C., and Strube, M. Multi-level annotation of linguistic data with MMAX2. In *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, S. Braun, K. Kohn, and J. Mukherjee, Eds. Peter Lang, Frankfurt a.M., Germany, 2006, pp. 197–214. (Cited on page 6)

[13] Ng, V. Unsupervised models for coreference resolution. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Morristown, NJ,

USA, 2008), Association for Computational Linguistics, pp. 640–649. (Cited on pages 6 and 12)

[14] Rahman, A., and Ng, V. Supervised models for coreference resolution. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (Morristown, NJ, USA, 2009), Association for Computational Linguistics, pp. 968–977. (Cited on page 6)

[15] Rauber, A., Merkl, D., and Dittenbach, M. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks 13*, 6 (nov 2002), 1331 – 1341. (Cited on page 15)

[16] Sang, E. F. T. K. Memory-based shallow parsing. *The Journal of Machine Learning Research* (2002). (Cited on page 12)

[17] Stoyanov, V., Cardie, C., N., G., Riloff, E., D., B., and D., B. Reconcile: A coreference resolution research platform. Tech. rep., Lawrence Livermore National Laboratory, 2009. (Cited on page 6)

[18] Tasdemir, K., and Merenyi, E. Exploiting data topology in visualization and clustering of self-organizing maps. *IEEE Transactions on Neural Networks 20*, 4 (2009), 549 –562. (Cited on page 18)

[19] Ultsch, A. U*-matrix: a tool to visualize clusters in high dimensional data. Tech. rep., DataBionics Research Lab, Department of Computer Science University of Marburg, 2003. (Cited on page 17)

[20] Versley, Y., Ponzetto, S. P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., and Moschitti, A. Bart: a modular toolkit for coreference resolution. In *HLT '08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies* (Morristown, NJ, USA, 2008), Association for Computational Linguistics, pp. 9–12. (Cited on page 6)

[21] Vesanto, J., Himberg, J., Alhoniemi, E., and Parhankangas, J. Self-organizing map in matlab: the som toolbox. In *Proceedings of the Matlab DSP Conference* (1999). (Cited on page 11)

[22] Witte, R., and Tang, T. Task-Dependent Visualization of Coreference Resolution Results. In *International Conference on Recent Advances in Natural Language Processing (RANLP 2007)* (Borovets, Bulgaria, September 27–29 2007). (Cited on page 6)

[23] Yianilos, P. N. Normalized forms for two common metrics. In *NEC Research Institute, Report 91-082-9027-1, 1991, Revision 7/7/2002. http://www.pnylab.com/pny* (1991), Cambridge University Press. (Cited on page 12)

Alle URLs wurden zuletzt am 01.11.2010 geprüft.