Universität Stuttgart

Institut für Maschinelle Sprachverarbeitung

Pfaffenwaldring 5b
70569 Stuttgart

Bachelorarbeit

Optimierung von Clustering von Wortverwendungsgraphen

Benjamin Tunc

Studiengang: Softwaretechnik - B.Sc.

1. Prüfer: Apl. Prof. Dr. Sabine Schulte im Walde

Betreuer: Dominik Schlechtweg

begonnen am: 15.05.2021

beendet am: 15.11.2021

Contents

Abstract

Algorithms for clustering of Word Usage Graphs are not optimal in terms of efficiency and often do not find the optimal clustering loss on larger graphs. Our aim in this paper is to find efficient ways to approximate the global minimum of a clustering loss function on three Word Usage Graphs data sets using correlation clustering and simulated annealing. Therefore we define 321 models with different initialization modifications, parameter combinations and stopping criterion and evaluate them in terms of loss, similarity to word sense description annotation, robustness and runtime. We evaluate different approaches and define efficient models with dynamic stopping criterion to find the lowest loss, which yield robust cluster solutions. We find that lowering the loss lead to better and clustering solutions.

1 Introduction

Word Usage Graphs (WUGs, McCarthy et al., 2016; Schlechtweg et al., 2021b) represent usages of a word as nodes in a graph which are connected by weighted edges representing semantic proximity. (Find examples in Figure 1.) WUGs are a convenient way to represent pairwise human semantic proximity judgments of word usages and then to infer word senses by clustering usages without the need for a priori word sense descriptions (Schlechtweg et al., 2020). Schlechtweg et al. (2020) recently introduced a version of correlation clustering (Bansal et al., 2004) using simulated annealing (Pincus, 1970) to find the optimal cluster structure on WUGs. However, the proposed algorithm is not optimal in terms of efficiency and often does not find the optimal clustering loss on larger graphs even after several iterations with brute-force settings. Our aim in this paper is to find efficient ways to approximate the global minimum of Schlechtweg et al. (2020)'s clustering loss function on three WUG data sets. For this, we test several initialization modifications, parameter combinations and stopping criteria. We further evaluate the reproducibility of the clustering solutions using robustness checks and compare them to independently obtained clusterings as external evaluation criterion. Using different approaches in initialization, we find that using a dependently initialized model deliver better results than independent initialized models. However, we find that pure dependent initialization does not lead to

optimal cluster solution, and that it always accompanied by additional random initialization. We compared repetitive models to non-repetitive models. They obtained not only better results, but also enabled the use of stopping criteria, saving up to 47% runtime. We find that the optimization of the loss leads to better cluster solutions and optimized models yield robust cluster solutions. Reproducibility across data sets is not fully given, but shows strong similarities.

2 Related Work

Human semantic proximity judgments have been proposed to compare word usages by researchers from multiple disciplines (Blank, 1997; Brown, 2008; Erk et al., 2009; 2013). Erk et al. (2013) provide the first in-depth study of semantic proximity judgments. McCarthy et al. (2016) represent this type of judgments within graphs, while Schlechtweg et al. (2021b) provide scalable annotation strategies for such graphs and create a large multilingual resource of human semantic proximity judgments represented in WUGs. While McCarthy et al. cluster the uses based on heuristics such as connected components, other cluster approaches have been proposed, including probabilistic modelling (Schlechtweg et al., 2021a) and correlation clustering (Schlechtweg et al., 2020; 2021b). Schlechtweg et al. define clustering as a discrete optimization problem. With simulated annealing they aim to find a clustering minimizing a loss function on edge weights derived from semantic proximity judgments.

3 Data

We utilize the annotated English, German and Swedish DWUG datasets (EN V1.0.0, DE V1.1.0, SV 1.0.0) (Schlechtweg et al., 2021b).[2] Each dataset contains a list of target words and a set of usages per target word from two time periods, t_1 and t_2 (we ignore the time allocation within the experiments). Each dataset includes 30-50 words each, with up to 200 usages per word. The word usages are combined into pairs as in (1) and (2) for the

[2]https://www.ims.uni-stuttgart.de/data/wugs

target word *plane* and annotated for their semantic proxmity on the DURel relatedness scale, which is illustrated in Table 1.

(1) Von Hassel replied that he had such faith in the **plane** that he had no hesitation about allowing his only son to become a Starfighter pilot.

(2) This point, where the rays pass through the perspective **plane**, is called the seat of their representation.

These judgments are then represented in a Word Usage Graph (WUG). WUGs are weighted, undirected graphs, which can be defined $\mathbf{G} = (\mathbf{U}, \mathbf{E}, \mathbf{W})$ with nodes $u \in U$ representing word usages and weights $w \in W$ representing the median of the human judgments for the corresponding pair of usages. Find examples in figure 1.

Human-annotated WUGs are often sparsely observed and noisy, which has different reasons. On the one hand, the number of edges increases quadratically with the number of nodes often resulting in magnitudes which are impossible to annotate. Hence, WUGs are often only partly annotated. On the other hand, usages can be ambiguous and unfamiliar to annotators leading to disagreements between annotators and acting as noise for the clustering approach.

3.1 Sense descriptions

An alternative approach to semantic proximity judgments are word sense description annoations (cf. Kilgarriff, 1998). For these, each usage is assigned to one of multiple word sense descriptions provided to annotators. The DWUG DE dataset was annotated with additional word sense descriptions (Schlechtweg et al., 2021b). For this, 24 words were randomly chosen and for each word 50 randomly sampled usages were annotated. We exclude all usages with at least one 'other' judgment and all usages where at least one annotator pair diverges on their judgments. The remaining usages are used to calculate the ARI values as described in more detail in Section 5.

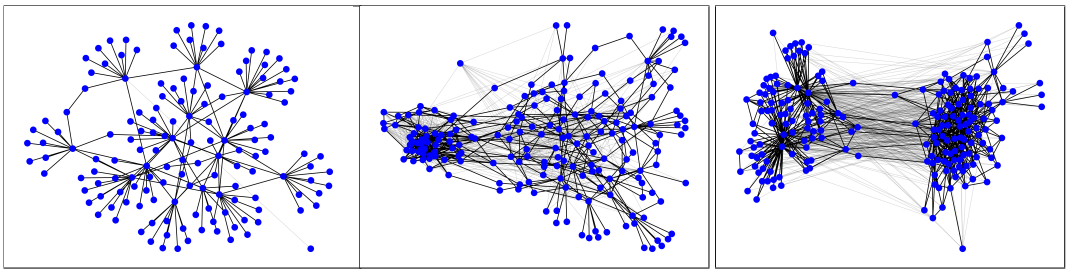

Figure 1: WUGs of German *Festspiel* (left), *Abgesang* (middle) and *zersetzen* (right) from the DWUG DE data set. Nodes represent usages of the respective target word. Edge weights represent the median of relatedness judgments between usages (**black**/graygray lines for **high**/graylow edge weights, i.e., weights ≥ 2.5/weights < 2.5).

	4: Identical
↑	3: Closely Related
|	2: Distantly Related
	1: Unrelated

Table 1: DURel relatedness scale defined in (Schlechtweg et al., 2018)

4 DWUG correlation clustering

Correlation Clustering describes a method for dividing the nodes of a weighted graph $\mathbf{G} = (\mathbf{U}, \mathbf{E}, \mathbf{W})$ into an optimal number of clusters (Bansal et al., 2004). In the most simple case, weights $W(e)$ on edges $e = (u, v) \in E$ are binary values $W(e) \in \{-1, 1\}$, i.e., either negative $(-)$ or positive $(+)$, depending on the similarity of the nodes u and v. Bansal et al. then try to minimize the sum of positive edge weights between different clusters ($W(e) = +$ and $u \notin C(v)$) and the sum of negative edges weights within clusters ($W(e) = -$ and $u \in C(v)$). Schlechtweg et al.'s DWUG correlation clustering varies from Bansal et al.'s most simple case, as edge weights are non-binary. For this, the weights $W(e)$ of all edges $e \in E$ in a WUG G are shifted to $W'(e) = W(e) - 2.5$ (e.g. a weight of 4 becomes 1.5). Those edges $e \in E$ with a weight $W'(e) \geq 0$ are referred to as **positive** edges P_E, while edges with weights $W'(e) < 0$ are called **negative** edges N_E. Let further

C be some clustering on U, $\phi_{E,C}$ be the set of positive edges **across** any of the clusters in clustering C and $\psi_{E,C}$ the set of negative edges **within** any of the clusters. We then search for a clustering C that minimizes $L(C)$:

$$L(C) = \sum_{e \in \phi_{E,C}} W'(e) + \sum_{e \in \psi_{E,C}} |W'(e)| \tag{3}$$

That is, the sum of positive edge weights between clusters and (absolute) negative edge weights within clusters is minimized ('loss').

4.1 Optimization: Simulated Annealing

Minimizing L is a discrete optimization problem which is NP-hard (Bansal et al., 2004). This is eased by the relatively low number of nodes (≤ 200). Hence, Schlechtweg et al. approximate the global optimum with Simulated Annealing (Pincus, 1970).[3] We define a temperature parameter T dependent on iteration i where $T(i) = max(T_0 * e^{-di}, T_{min})$ with $0 \leq T \leq 1$, and T_0 being the initial temperature, T_{min} being the minimum temperature and d being the rate of exponential decay. We choose the default parameter values provided by mlrose: $T_0 = 1.0$, $T_{min} = 0.001$ and $d = 0.005$. As displayed in Algorithm 1, in every iteration we calculate $T(i)$ and choose a random neighbor state $nghbr$ based on the current state $state$. If the loss $L(nghbr)$ is lower than $L(state)$, or if $L(nghbr)$ is higher than $L(state)$ and $prob$ is sufficiently high, then the existing state is replaced by the random neighbor. For example, let the loss of the current state $L(state) = 10$, the loss of the random neighbor $L(nghbr) = 12$ and the algorithm be in the twentieth iteration (yielding $T(20) = 0.905$ and $prob = 0.110$). Then, there is a probability of roughly 0.11 that $nghbr$ is chosen. With decreasing T, $prob$ also decreases, i.e., in later iterations the change to a state with higher loss occurs less often. The algorithm runs until either the T drops to 0 or the maximum number of iterations or maximum number of attempts is reached.

[3] We use Schlechtweg et al. (2021b)'s code: https://www.ims.uni-stuttgart.de/data/wugs relying on mlrose (Hayes, 2019).

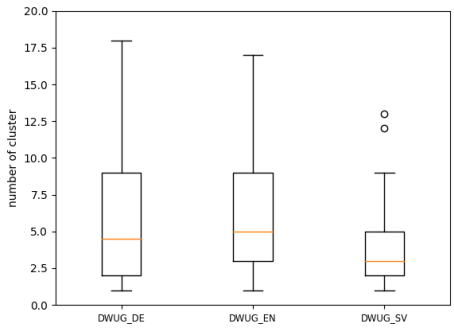

Figure 2: Number of clusters per dataset. Two words (*tip* EN, 63 clusters; *medium* SV, 23 clusters) exceed the value of 20 clusters and are therefore not visible.

4.2 Parameters

Schlechtweg et al.'s clustering approach has multiple parameters which we describe in the following section.

Maximum number of clusters The search space of Simulated Annealing's neighbor function is bounded by k, the maximum number of clusters (senses) it can assign. Schlechtweg et al. iterate over different values for $0 \leq k \leq s$ in order to reduce the search space for each value of k, in contrast to choose $k = s$ for each iteration. We iterate over $s \in \{5, 7, 10, 15, 20\}$. As we can see in Figure 2 these values capture the large majority of cluster numbers from the optimal cluster solutions provided by Schlechtweg et al.

Maximum number of attempts and iterations for simulated annealing Two of the parameters of mlrose's Simulated Annealing have influence on performance and runtime. Maximum attempts ($maxA$) is the maximum number of attempts to switch to a neighbor state and maximum iterations ($maxI$) is the maximum number of iterations the algorithm will run, cf. Algorithm 1. For both parameters an increase means longer runtime (ignoring the special case that the algorithm breaks at the smaller $maxI$ or $maxA$), while either a

better neighbor state can be found (condition $delta_e > 0$) or a worse state can be found (condition $random < prob$). However, as $prob$ decreases with a longer runtime, later changes to states have higher probabilities to be better. For our experiments we have included the following parameter combinations based on parameters chosen by Schlechtweg et al. (2020) and initial experiments: $maxA/maxI \in \{100/10000, 100/20000, 500/10000, 1000/10000, 1000/20000, 5000/10000, 5000/20000\}$

Number of repetitions Schlechtweg et al. repeat the whole clustering procedure only once. We change this in our approach, because we noticed that one repetition often does not find the optimal clustering solution. We repeat the optimization up to 10 times and save the best result from all previous repetitions.

Initialization For each value of k (maximum number of clusters) Schlechtweg et al. perform one clustering with a random initialization, and a second clustering initialized with a heuristically chosen clustering solution derived from connected components on edge weights above the clustering threshold.[4] In addition to this approach we exchange the heuristically chosen initialization for an initialization with the best solution found in all previous repetitions. The first repetition (where no previous solution is available) is still initialized heuristically. Note that in the first approach each repetition is *independent* from the previous ones, while in the second approach later repetitions are *dependent* on earlier ones. We assume that by initializing with a good solution the probability of finding the global minimum is higher than initializing with a random state. However, we additionally initialize with a random state to avoid being stuck on a local minimum.

Stopping criteria At each repetition r the optimal loss L_r found in all previous repetitions $s \leq r$ is smaller or equal to all L_s. We also find that L_r converges, as exemplified in Figure 3. We define several stopping criteria which aim to stop the algorithm when it converged or nearly converged. This makes it possible to achieve similarly good results, which may not be optimal, but significantly reduce the runtime. Note that we extract model

[4] In case that the heuristic clustering solution has a cluster number $n > k$ we set k to n.

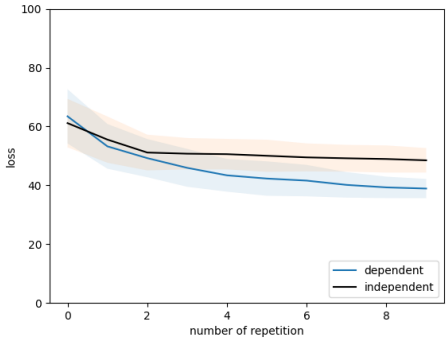

Figure 3: Development of loss per repetition for German *Knotenpunkt* with model $s = 20$, $maxA = 500$, $maxI = 10000$, independent and dependent initialization.

results with stopping criteria post hoc from the full 10 repetitions. Hence, all results are dependent by being computed on the same 10 values.

Fixed number of repetitions This static approach has a fixed number of repetitions r after which the algorithm stops. Within our experiments we use $r = 5$ and $r = 10$.

Comparison with last repetition ($\mathbf{r = l1}$) This approach compares the current optimal loss with the optimal loss of the previous repetition. The criterion comes into effect if the improvement is less than 2%.

Comparison with the last three repetitions ($\mathbf{r = l3}$) This approach compares the current optimal loss with the average optimal loss of the last three repetitions. The criterion comes into effect when the improvement is less than 3%. We assume that the average is more robust to saddle points.

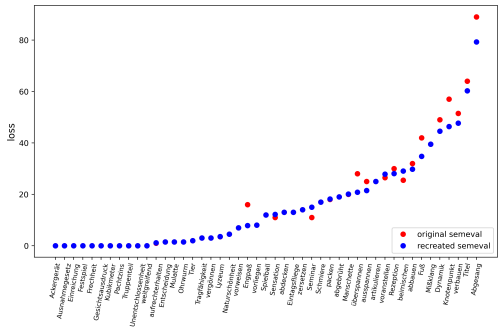

Figure 4: loss per word of *semeval* clusterings provided by Schlechtweg et al. and our model $s = 20$, $maxA = 2000$, $maxI = 50000$, independent initialization and $r = 5$.

4.3 Baseline

We compare our results to the *semeval* clusterings provided by Schlechtweg et al.. We call this model *semeval*. We also reproduce their results with similar parameters. However, they used a sophisticated combination and iteration over parameters which we can only approximate: We take $s = 20$, $maxA = 2000$, $maxI = 50000$, independent initialization and $r = 5$. This model achieved similar results in our initial experiments to Schlechtweg et al. as can be seen in Figure 4. In total, we have defined 321 models to evaluate. Each model is defined by a specific parameter combination, i.e., of a maximum number of clusters s, $maxA$, $maxI$, independent or dependent initialization and a stopping criterion.

5 Evaluation

We compare the models with respect to the following metrics:

Loss Loss is the value of $L(C)$, defined in Section 4.

Runtime The number of seconds a model runs in total.

Algorithm 1 Simulated Annealing[5]

$state \leftarrow initial$
while $attempts < maxA$ and $i < maxI$ **do**
  $temp \leftarrow T(i)$
  $i \leftarrow i + 1$
  if $temp = 0$ **then**
    break
  else
    $delta_e \leftarrow L(nghbr) - L(state)$
    $prob \leftarrow exp(delta_e/temp)$
    $random \leftarrow random(0, 1)$
    if $delta_e > 0$ or $random < prob$ **then**
      $state \leftarrow nghbr$
      $attempts \leftarrow 0$
    else
      $attempts \leftarrow attempts + 1$
    end if
  end if
end while

Adjusted Rand Index For each cluster solution we calculate the Adjusted Rand Index (ARI)(Hubert and Arabie, 1985) to the human-annotated sense descriptions as defined in Section 3.[6] This value is an external evaluation criterion to determine the cluster quality and is bounded between 0.0 and 1.0. If $ARI = 1.0$, the clusters are identical.

Robustness For each cluster solution we calculate a robustness value by averaging all ARI scores of that solution to the final cluster solutions of the 10 experiment runs (see below). The Robustness is bounded between 0.0 and 1.0.

[6]We use the sklearn implementation (Pedregosa et al., 2011).

6 Experiments

We run each model 10 times and take the average values for each metric per word to eliminate inaccuracies. We use the median values to aggregate over all words, as distributions of values are often strongly skewed. We analyze model results on German words, but also discuss the results on the other data sets.

Which model finds the lowest loss? Table 2 shows the 16 models with the lowest median loss of 8.0 over all German words. For better overview we summarize rows over stopping criteria. We see that all models in Table 2 are dependently initialized. The best independent model is in the upper 21th percentile with a median loss of 10.8. Schlechtweg et al. (2020)'s *semeval* clustering has a median loss of 11.0. Most models have relatively good results and are at a median loss of 12.0 to 13.0, as can be seen in Figure 5. All models in the lower 9th percentile are models with $maxA = 100$ with different stopping criteria and s values. Most of these models are initialized independently, however, there are also some dependent models amongst them. Comparing the frequency of the stopping criteria in Table 2, we notice that $r = 10$ (11 of 13) and $r = l3$ (3 of 13) are the most frequent in the list. Since $r = 10$ always runs through the maximum number of repetitions, it is not possible for other stopping criteria with similar parameters to achieve better results. In some cases $r = l3$ has achieved equally good results, which saved on average 47% of the runtime. Regarding the top models, the following pattern of dominating parameters can be derived: $s \in 10, 15, 20$, $maxA \in \{500, 1000, 5000\}$, $maxI \in \{10000, 20000\}$ and dependent initialization.

Are repetitions useful? We run a model ($S = 20$, $maxA = 5000$, $maxI = 50000$, $r = 1$) with significantly increased parameters but only one repetition and compare this model with a repetitive model with a similar runtime ($S = 20$, $maxA = 500$, $maxI = 10000$, independently initialized, $r = l3$,). We choose a model with the same S to ensure fairness. The model with increased parameters has a median loss of 13.0 and a median runtime of 29s. The repetitive model has a median loss of 9.55 and a runtime of 22s. Similar results were also obtained with different s values. Even with less runtime, the

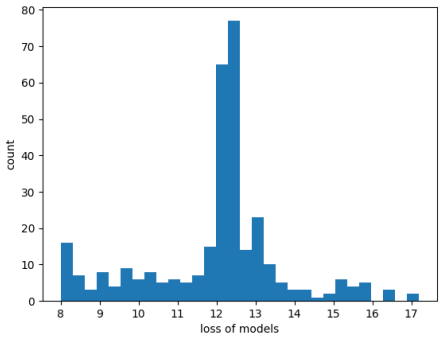

Figure 5: Frequency of all median loss values.

repetitive model beats the non-iterative model. Furthermore, the possible use of dynamic stopping criteria on iterative models is advantageous for the runtime.

Which model is most efficient? We select all models with a median loss below or equal to 10.0 (upper 16th percentile) and compare them in terms of their runtime. The model $s = 10$, $maxA = 500$, $maxI = 10000$, dependently initialized and $r = l1$ with loss of 9.35 beats all other 49 models in terms of runtime (7s). The model is overall in the upper 11th percentile in terms of runtime and has low median loss (upper 13th percentile). However, note that there are also efficient models reaching the lowest loss in Table 2, e.g. $s = 20$, $maxA = 500$, $maxI = 10000$, independtly initialized $r = l3$ has a median loss 8.0 and a runtime of 17s. The most inefficient model is the recreated semeval model: $s = 20$, $maxA = 2000$, $maxI = 50000$, independent initialization and $r = 5$, with a runtime of 148s it is the slowest of all models.

Which stopping criterion is best-performing? We compare stopping criteria over all models in Table 4. It can be seen that $r = l1$ provides the lowest runtime, but the highest loss. The criteria with the most repetition $r = 10$ has the lowest loss and (as expected) the highest runtime. $r = l3$ reaches lower loss (1.0 more than $r = 10$) and has a comparatively low runtime (half of $r = 10$). As an example, if this stopping criterion $r = 10$ has reached

s	maxA	maxI	init	r	loss	runtime	ARI	robust
10	500	10000	depen.	10/5/l3	**8.0**	30/15/**17**	.72/.70/.71	.97/.96/.96
15	500	10000	depen.	10	**8.0**	36	.71	.98
20	500	10000	depen.	10	**8.0**	43	**.73**	.98
10	500	20000	depen.	10/l3	**8.0**	51/27	.72/.72	.98/.97
15	500	20000	depen.	10	**8.0**	51	**.73**	.98
20	500	20000	depen.	10	**8.0**	62	**.73**	**.99**
10	1000	10000	depen.	10/l3	**8.0**	31/**17**	**.73**/.72	.98/.97
15	1000	20000	depen.	10	**8.0**	64	**.73**	.98
15	5000	20000	depen.	10	**8.0**	32	.72	.98

Table 2: Overview of all models with lowest median loss of 8.0 over German target words.

the optimal cluster solution in the first repetition, the 9 further repetitions are superfluous. Due to this static behavior, this stopping criterion is comparatively inefficient. As can be seen in Figure 3, the development of loss per repetition converges to a limit, which makes a dynamic stopping criterion more efficient. Even the models with lowest median loss in Table 2 only differ about 7% in terms of loss to their respective model with $r = 10$ with a reduction of the runtime about 47%. $r = 5$'s runtime is half of $r = 10$ and has in average a median loss of 11.6, meaning $r = 10$ used half of the runtime to only improve about 2.0 in terms of loss. Since $r = 5$ is still a static model, it can be inefficient in less complex words.

Does lower loss mean higher robustness? In Figure 6 we see a scatter plot of loss vs. robustness values of all cluster solutions. We notice a clustering of the points in the upper left corner (low loss and high robustness). Furthermore, we calculate the Spearman correlation between the two variables yielding a strong negative correlation of -0.73. This means that lower loss implies higher robustness, i.e., more reproducible clustering solutions. This is also confirmed by the models with the lowest median loss in Table 2 with high robustness values between 0.96 and 0.99.

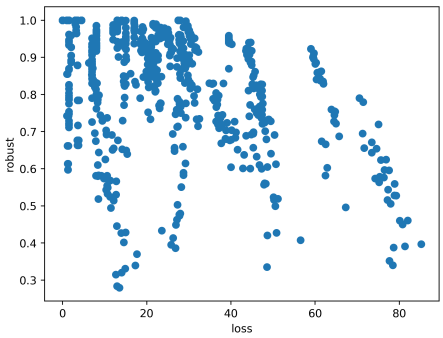

Figure 6: Loss vs. robustness values of all cluster solutions for German words.

Which model gives the highest ARI? The highest median ARI that occurring in the experiments is .73 and was achieved by 16 models. Within the 16 models, every $svalue$ occurs, with 20 being the most common. Furthermore, every combination of $maxA$ and $maxT$ is represented, except the two with $maxA = 100$. All models are dependent initialized and have the stopping criterion $r = 10$. Please note, all models from table 2 have an approximately good ARI value, so we assume that the optimization of the loss also leads to an optimization of the ARI. Of the 16 models with the highest ARI , 5 are within the models with the lowest median loss , 11 are in the upper 11th percentile in terms of loss and two are in the upper 20th. We further investigate the behavior of ARI to loss and calculate the Spearman correlation between all loss and all ARIs values for each word. Almost all correlations are negative and show moderate correlation (-0.45) at the median with one word (*Titel*) deviates strongly from the other correlations and has a positive correlation. Therefore, it can be concluded that minimizing the loss leads to a better cluster solution. However, only for 19 of 24 German words with sense descriptions, the cluster solution with the highest ARI is also the solution with the lowest loss. As described in more detail in Section 3, WUGs are often sparsely observed and noisy, which can lead to finer clusterings. As an example, we consider the plots of *Titel* produced by Schlechtweg et al.. We see that two nodes have been classified in different clusters, even though they both have the same sense description with full matching of annotators.

init.	loss	runtime	ARI	robustness
indepen.	12.7	28	.62	.92
depen.	**10.6**	**27**	**.70**	**.96**

Table 3: Average of metrics of independently and dependently initialized models with static stopping criteria over German words.

Does dependent initialization beat independent initialization? Table 3 shows the average values of all medians of models with independent and dependent initialization and static stopping criteria ($r = 5$ and $r = 10$).[7] As can be seen, dependent initialization gives better results than independent initialization for every metric. Furthermore, all models with the lowest median loss in Table 2 are dependently initialized. We investigate whether this behavior is caused by the connected components and repeat for the models $maxA = 500$ and $maxI = 10000$ all models with static stopping criteria and independent initialization where we replace the initialization with connected components by another random initialization. On average, these models have a median of 11.7, which shows a difference of 0.5 to the models initialized with connected components. Furthermore, we repeat all models with $maxA = 500$ and $maxI = 10000$ and dependently initialized without additional random initialization and compare them to the models with the same parameter combination from our experiments. In the average of the medians, the models without random initialization have a loss of 24.0. Compared with the loss from their respective counterpart (9.8), this value is significantly increased. We assume that these models are stuck on a local minimum for complex words, since comparatively good results could be obtained for less complex words, e.g. model $S = 10$, dependently initialized, $r = 10$ (without additional random initialization) has a loss of 79.9 for *Knotenpunkt*, while the same model with additional random initialization has a loss of 41.2.

[7] We only use the stopping criteria $r = 10$ and $r = 5$, since the other stopping criteria have different numbers of repetitions.

r	ARI	loss	robust	runtime
10	**.72**	**9.5**	**.97**	64.6
5	.69	11.6	.95	32.5
l1	.68	12.0	.94	**14.7**
l3	.70	10.5	.95	35.1

Table 4: Average of metrics of stopping criteria over German words.

Which are the best parameters for simulated annealing? Table 5 displays the average metrics per parameter combination of $maxA$ and $maxI$. Due to the relatively low number of maximum attempts, models with $maxA = 100$ stopped too early, which is why they scored worse on average for loss, ARI and robustness. Amongst the top models in Table 2 all parameter combinations except $maxA = 100$ are represented. Although we doubled $maxA$ for parameter combinations with $maxA \leq 500$, it does not affect metric noticeable and seem to be influenced significantly more by their $maxI$ value. A parameter combination of $maxA = 500$ and $maxI = 10000$ is sufficient to achieve good results.

Are results consistent across data sets? We run the same experiments for both English and Swedish words. For the Swedish words, the median losses of all models were between 2.5 and 3.6, so all models found a relatively low loss. The lowest loss of 2.5 was achieved by 9 independent models and 29 dependent models and the stopping criterion $r = l3$ is next to $r = 10$ are the most frequent amongst them. If we compare the stopping criteria, we notice that $r = 10$ (again) achieves the lowest average loss, but $r = i5$ and $r = l3$ come very close with a difference of 0.1. The s values of the models with the lowest loss are also $s \leq 10$. Also for the Swedish words the development of the losses per repetition converges to a limit value, but reaches it after only a few repetitions, which makes the difference in average loss between $r = i5$ and $r = 10$ less noticeable. For the English words we could not reproduce this. The lowest loss of 14.0 was achieved by independent models, the best dependent model has a loss of 14.9. We note that for some words in the English data set,

maxA	maxI	loss	runtime	ARI	robust
100	10000	12.9	**14**	.63	.89
100	20000	12.8	15	.63	.89
500	10000	**11.0**	25	.67	**.96**
500	20000	11.1	42	.67	**.96**
1000	10000	11.2	25	.67	**.96**
1000	20000	11.3	46	**.68**	**.96**
5000	10000	11.1	26	**.68**	**.96**
5000	20000	11.4	46	**.68**	**.96**

Table 5: Average of median results of different $maxI$ and $maxA$ combinations over German words.

the cluster solutions of a model can vary greatly, resulting in comparatively low robustness values. For example, the word plane has robustness values between 0.50 and 0.68. Looking at the average values for depend initialization and independent initialization both have an average loss of 16.5. For the stopping criteria, it is the same behavior as for the Spanish and German data sets. Reproducibility across data sets is not fully given, but shows strong similarities.

7 Conclusion

Within this thesis we have done large-scale experiments to find efficient ways to approximate the global minimum of Schlechtweg et al. (2021b)'s clustering loss function on three WUG data sets. Through several initialization modifications, parameter combinations and stopping criteria, we defined 321 different models that we evaluated for loss, ARI, runtime and robustness. According to the Results, we provided a pattern for models to obtain the lowest median loss and have shown that the optimization of the loss leads to cluster solutions that are closer to the sense descriptions and therefore can be consider as

better. The results lead to the conclusion that a correlation exists between ARI and loss, as well as between loss and robustness, which we have further analyzed. Furthermore, we have shown that a repetitive approach leads to lower loss values than the approach offered by Schlechtweg et al. (2021b) and also allows the use of different stopping criteria to reduce runtime significant. We have shown that best models produce robust solutions and thus have high reproducibility. We also discuss the differences between the results of the data sets and find that the reproducibility between the data sets is not complete, but shows strong similarities.

Future work involves analysis of simulated annealing parameters with variation in the temperature function, experiments using other optimization algorithms (e.g. random hill climb) or other stopping criteria, or comparison of correlation clustering with other clustering algorithms (Biemann, 2006; McCarthy et al., 2016; Schlechtweg et al., 2021a).

References

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004. doi: 10.1023/B:MACH.0000033116.57574.95.

Chris Biemann. Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, page 73–80, USA, 2006. Association for Computational Linguistics.

Andreas Blank. *Prinzipien des lexikalischen Bedeutungswandels am Beispiel der romanischen Sprachen*. Niemeyer, Tübingen, 1997.

Susan Windisch Brown. Choosing sense distinctions for WSD: Psycholinguistic evidence. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 249–252, Stroudsburg, PA, USA, 2008.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. Investigations on word senses and word usages. 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: Volume 1, pages 10–18, Stroudsburg, PA, USA, 2009.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. Measuring word meaning in context. *Computational Linguistics*, 39(3):511–554, 2013.

Genevieve Hayes. mlrose: Machine Learning, Randomized Optimization and SEarch package for Python. `https://github.com/gkhayes/mlrose`, 2019. Accessed: May 22, 2020.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2: 193–218, 1985.

Adam Kilgarriff. Itri-98-09 senseval: An exercise in evaluating word sense disambiguation programs. 1998.

Diana McCarthy, Marianna Apidianaki, and Katrin Erk. Word sense clustering and clusterability. *Computational Linguistics*, 42(2):245–275, 2016.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Martin Pincus. A monte carlo method for the approximate solution of certain types of constrained optimization problems. *Operations Research*, 18(6):1225–1228, 1970. doi: 10.1287/opre.18.6.1225.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. Diachronic Usage Relatedness (DURel): A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 169–174, New Orleans, Louisiana, 2018. URL `https://www.aclweb.org/anthology/N18-2027/`.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain, 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.semeval-1.1/.

Dominik Schlechtweg, Enrique Castaneda, Jonas Kuhn, and Sabine Schulte im Walde. Modeling sense structure in word usage graphs with the weighted stochastic block model. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 241–251, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.starsem-1.23. URL https://aclanthology.org/2021.starsem-1.23.

Dominik Schlechtweg, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. DWUG: A large Resource of Diachronic Word Usage Graphs in Four Languages. *CoRR*, abs/2104.08540, 2021b. URL https://arxiv.org/abs/2104.08540.

A Additional abstract in German

Algorithmen für das Clustering von Wortverwendungsgraphen sind im Hinblick auf ihre Effizienz nicht optimal und finden oft nicht den optimalen Clustering-Loss bei größeren Graphen. Unser Ziel in diesem Arbeit ist es, effiziente Wege zu finden, um das globale Minimum einer Clustering-Lossfunktion auf drei Wortverwendungsgraphen-Datensätzen mit Hilfe von Korrelationsclustering und Simulated Annealing zu approximieren. Zu diesem Zweck definieren wir 321 Modelle mit unterschiedlichen Initialisierungsmodifikationen, Parameterkombinationen und Abbruchkriterien und evaluieren sie in Bezug auf Loss, Ähnlichkeit mit Word Sense Description, Robustheit und Laufzeit. Wir evaluieren verschiedene Ansätze und definieren effiziente Modelle mit dynamischem Abbruchkriterium, um den geringsten Loss zu finden und zeigen dass diese zu robusten Clusterlösungen führen. Wir stellen fest, dass eine Verringerung des Verlusts zu besseren und robusteren Clusterlösungen führt.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Druck-Exemplaren überein.

Datum und Unterschrift: 15.11.2021 B.Tunc

Declaration

I hereby declare that the work presented in this thesis is entirely my own. I did not use any other sources and references that the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted hard copies.

Date and Signature: 15.11.21 B.Tunc