Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)

Label Propagation of Polarity Lexica on Word Vectors

Harald Koppen
cellent GmbH
Lehrer-Wirth-Str. 2
81829 München
haraldkoppen@gmail.com

Ritavan
Surplus Digital GmbH
Georg-Muche-Straße 5
80807 München
ritavan.saice@gmail.com

Abstract

The Semi-supervised learning (SSL) is an important research area in machine learning where both labeled and unlabeled data is used to build a model. One of the big advantages of semi-supervised methods is that they are transparent and easy to comprehend for humans, unlike most deep learning techniques which are black box. In this paper, we design a graph-based semi-supervised learning framework to detect sentiment polarity in word vectors trained on a German corpus. We study theoretical aspects of the task, empirically analyze a seminal label propagation algorithm (Zhu and Ghahramani, 2002) and suggest variants to improve classification performance. Additionally, we review the literature of graph construction for SSL and propose new methods to avoid hubs, i.e., vertices of high degree, which are harmful as outlined by Ozaki et al. (2011).

1 Introduction

Among the most ubiquitous techniques for label enrichment and transfer learning in sentiment analysis, in particular for classification tasks, are sentiment lexica and word vectors. The use of such lexica is a classical approach which has been used for several decades before the advent of deep learning (Taboada et al., 2011). The training of word vectors from large unlabeled text corpora is a comparatively more recent method dating back to the seminal paper by Mikolov et al. (2013).

For sentiment analysis, it is common to focus on supervised methods (Gamon, 2004; Matsumoto et al., 2005; Pang et al., 2002; dos Santos and Gatti, 2014). Usually, large unlabeled text corpora are easily available, whereas labeled lexica are harder to come by and often involve exorbitant labeling costs. Thus, given an unlabeled dataset at the outset, this approach is expensive as it takes both time and labor to annotate a sufficiently large training set. Typically, word vectors have a vocabulary of size $O(10^6)$ (Mikolov et al., 2013) while lexica contain $O(10^4)$ (Waltinger, 2010a) words, thus resulting in a poor ratio of labeled to unlabeled points.

In recent years, semi-supervised learning (SSL) methods, particularly graph-based approaches based on label propagation (Zhu and Ghahramani, 2002) attracted attention (Goldberg and Zhu, 2006; Rao and Ravichandran, 2009; Ren et al., 2012). As a consequence, graph construction for these methods emerged as a relevant field of study (Ozaki et al., 2011; de Sousa et al., 2013; Vega-Oliveros et al., 2014) as well as approaches minimizing a cost function derived from such a graph (Ravi and Diao, 2016). Note that label propagation and its variations are equivalent to certain minimization problems (Bengio et al., 2006).

Giulianelli (2017) used SSL on a word embedding obtained via a layer of a long short term memory (LSTM) recurrent network instead of using word vectors. However, training an LSTM is a supervised task, i.e. the method requires a large amount of labeled data in the first place, which defeats the purpose and is going against the main motivation behind SSL techniques.

A major challenge with high dimensional data is the curse of dimensionality, a well-known phenomenon particularly affecting methods based on nearest neighbour graphs. Radovanović et al. (2010) and subsequently Ozaki et al. (2011) showed that hubs, i.e. vertices of high degree, have a negative effect on classification results due to the fact that they are among the nearest neighbours of a large subset of the dataset.

We introduce the k nearest neighbor (kNN) graph, consider different variants of it and propose a trimming and a normalization procedure in order to combat hubs.

2 Contributions

To the best of our knowledge, there is no previous work carrying out a detailed theoretical and empirical study of SSL as described above, that is label propagation of a German sentiment lexicon on word vectors trained on a German corpus.

Our contributions are as follows:

- A study of theoretical challenges of label propagation on a polarity lexicon of word vectors.
- Benchmarking the performance of label propagation on different word vector models of varying dimensionality, including contextual language models.
- Extensive experiments to study the performance of label propagation empirically with a variety of parameter configurations and graph construction techniques.
- Proposition of 2 new methods to avoid the negative effect of hubs during label propagation.

The rest of this paper is organized in the following way. We introduce the SSL setting, label propagation and its problems in section 3. Our new methods for graph regularization are explained in section 4. Further motivation, analysis of the dataset used, the set-up and the results of our experiments are given in section 5. We conclude with section 6.

3 Graph-based SSL

We begin with a definition of SSL, then define the similarity function. Afterwards, we move on to graph construction and label propagation before discussing the challenges faced by these methods.

3.1 Similarity and Semi-Supervised Learning

Assuming the data is already given as a finite set of points in $\mathbb{R}^d$, $d \in \mathbb{N}$, let $l \in \mathbb{N}$ denote the number of labeled points, $u \in \mathbb{N}$ the number of unlabeled points and $n = l + u$ the total number of points. We are considering $\mathscr{L} = \{x_1, \ldots, x_l\} \subseteq \mathbb{R}^d$, the set of labeled points, and $\mathscr{U} = \{x_{l+1}, \ldots, x_n\} \subseteq \mathbb{R}^d$, the set of unlabeled points, where $x_i \neq x_j$ for every $i \neq j$, i.e. the points are pairwise distinct. The label of x_i is denoted by $y_i \in \{0, \ldots, \rho\}$, $\rho \in \mathbb{N}$. In this paper, we study binary classification, i.e. $\rho = 1$. Given $\{y_1, \ldots, y_l\}$, the goal of SSL is to predict $\{y_{l+1}, \ldots, y_n\}$ as accurately as possible.

The similarity function is a map

$$\sigma : \mathscr{L} \cup \mathscr{U} \times \mathscr{L} \cup \mathscr{U} \longrightarrow \mathbb{R}^+, (x, x') \mapsto \sigma(x, x'),$$

for instance

$$\sigma_\gamma(x, x') = f_\gamma(x - x'),$$

where

$$f_\gamma : \mathbb{R}^d \longrightarrow \mathbb{R}^+, x \mapsto e^{-\frac{\|x\|_2^2}{2\gamma^2}}$$

denotes the radial basis function and $\gamma > 0$.

Another example makes use of the k nearest neighbors of x in $\mathscr{L} \cup \mathscr{U}$, defined as follows. Let $k \in \{1, \ldots, n-1\}$, $x \in \mathbb{R}^d$ and $x_{(1)}, \ldots, x_{(n)}$ be a reordering of $\mathscr{L} \cup \mathscr{U}$ such that

$$\left\|x_{(1)} - x\right\|_2 \leq \cdots \leq \left\|x_{(n)} - x\right\|_2.$$

Then the k nearest neighbors of x in $\mathscr{L} \cup \mathscr{U}$ are

$$k\text{NN}(x, \mathscr{L} \cup \mathscr{U}) = \{x_{(1)}, \ldots, x_{(k)}\}.$$

Now, we can define

$$\sigma_k(x, x') = \begin{cases} 1 & x' \in k\text{NN}(x, \mathscr{L} \cup \mathscr{U}) \\ 0 & \text{otherwise} \end{cases}. \qquad (*)$$

Note that for every $i \in \{1, \ldots, n\}$ and every k we have that $x_i \in k\text{NN}(x_i, \mathscr{L} \cup \mathscr{U})$. To avoid this, one can define $k\text{NN}(x, \mathscr{L} \cup \mathscr{U}) = \{x_{(2)}, \ldots, x_{(k+1)}\}$. Including the distance of x and x' is possible by using

$$\sigma_{k,\gamma}(x, x') = \begin{cases} f_\gamma(x - x') & x' \in k\text{NN}(x, \mathscr{L} \cup \mathscr{U}) \\ 0 & \text{otherwise} \end{cases}.$$

3.2 Construction of the Underlying Graph

The vertices of the underlying graph are given by $\mathscr{L} \cup \mathscr{U}$. Consider the adjacency matrix $A \in \mathbb{R}^{n \times n}$ which is derived from the similarity matrix defined as $W = (\sigma(x_i, x_j))_{1 \leq i,j \leq n}$.

The easiest choice for A is W itself, where $\sigma = \sigma_\gamma$ yields a dense, undirected and weighted graph. As A is usually heavily involved in the classification of $\mathscr{U}$ it is desirable to use a sparse matrix to save computation time. In particular, a sparse adjacency matrix results in higher classification accuracy as noise and spurious relationships are reduced (Zhu, 2008; Ozaki et al., 2011).

Taking $\sigma = \sigma_k$ leads to a sparse, directed and unweighted graph, $\sigma = \sigma_{k,\gamma}$ to a sparse, directed and weighted graph known as a kNN graph. Usually, it is transformed into an undirected graph by choosing the adjacency matrix

$$W_{\max} = (\max(\sigma(x_i, x_j)), \sigma(x_j, x_i)))_{1 \leq i,j \leq n}.$$

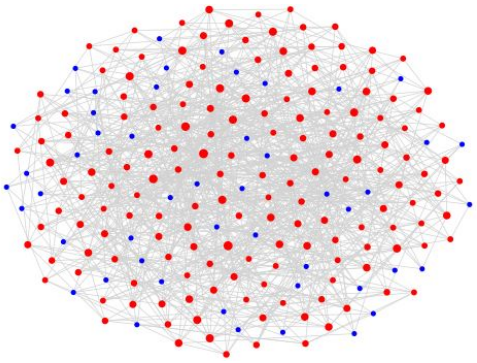

Figure 1: 5NN graph on UCI glass data set, where vertices with degree larger than 5 are drawn red and accordingly bigger.

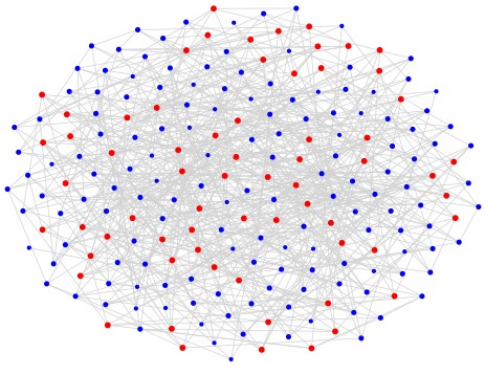

Figure 2: Trimmed version of Figure 1, $\alpha = 5$.

Ozaki et al. (2011) study the mutual kNN graph which is given by the adjacency matrix

$$W_{\min} = (\min(\sigma(x_i,x_j)),\sigma(x_j,x_i)))_{1\le i,j\le n}.$$

Note that $W_{\max}$ and particularly $W_{\min}$ may yield to a disconnected graph, harming the classification if there are connected components with few or absolutely no labeled points.

In any case, the use of σ_k results in self-loops as $\sigma_k(x_i,x_i) = 1$ for every $i \in \{1,\dots,n\}$. These can be removed by using the modified version of kNN mentioned below $(*)$. Note further that $\sigma_{k,\gamma}(x,x) = 0$ for every $x \in \mathbb{R}^d$, i.e. for fixed $i \in \{1,\dots,n\}$ there are not k, but $k-1$ non-zero entries in $(\sigma_{k,\gamma}(x_i,x_j))_{1\le j\le n}$. Again, the modified version of kNN prevents this behaviour.

3.3 Label Propagation

For the moment, let us assume $y_i \in \{-1,1\}$, i.e. we replace the label 0 by -1. Given an adjacency matrix A, the algorithm is given as follows (Bengio et al., 2006).

Algorithm 1 Label Propagation

Compute A
Compute diagonal D by $D_{ii} \leftarrow \sum_{j=1}^{n} A_{ij}$
Initialize $Y^{(0)} \leftarrow (y_1,\dots,y_l,0,\dots,0)$
Iterate
1. $Y^{(t+1)} \leftarrow D^{-1}AY^{(t)}$
2. $Y_i^{(t+1)} \leftarrow y_i$ for $1 \le i \le l$

until convergence criterion is satisfied
Denote the result by $Y^{(\infty)}$
Set $y_i = \operatorname{sgn}(Y_i^{(\infty)})$

Consequently, the algorithm propagates the information along the edges of the underlying graph, typically until an equilibrium state is reached. The nodes initially labeled serve as the source of information.

A classical assumption in SSL is the *cluster assumption*: if points are in the same cluster, they are likely to be of the same class (Chapelle et al., 2009). Of course, for high-dimensional data, it is hard to check if this assumption is fulfilled, especially given that only a small proportion of the data is labeled. Strictly speaking, this problem should be overcome by the word embedding algorithm, not the label propagation algorithm.

4 Improvements to the Graph

Let us now consider the undirected unweighted kNN graph without self-loops, that is, the graph $G = (\mathscr{L} \cup \mathscr{U}, W_{\max})$ with similarity function σ_k using the modifed version of kNN.

4.1 ε-Sparsification

Let $\varepsilon > 0$. The ε-sparsification of G is the graph G^{ε} which is obtained by deleting every edge $\{x_i,x_j\}$ in G where $\left\|x_i - x_j\right\|_2 > \varepsilon$. Therefore, using G^{ε} instead of G reduces the influence of outliers on the classification.

4.2 Edge Normalization

Firstly, we propose to transform G into a weighted graph G^n by performing *edge normalization*, i.e. by assigning every edge $\{u,v\}$ in G the weight

$$w_{u,v} = (\deg_G(u) + \deg_G(v))^{-1}.$$

Note that $w_{u,v}$ is small if u and v have high degree and vice versa, thus counterbalancing the high amount of edges between vertices with high degree.

Let $N_G(u)$ denote the set of neighbors of u in G.

For every vertex $u \in G$, we have

$$
\begin{aligned}
0 &< \sum_{v \in N_G(u)} w_{u,v} \\
&\leq \sum_{v \in N_G(u)} (\deg_G(u) + \min_{v \in N_G(u)} \deg_G(v))^{-1} \\
&= \frac{\deg_G(u)}{\deg_G(u) + \min_{v \in N_G(u)} \deg_G(v)} < 1,
\end{aligned}
$$

i.e. the weighted degree in G^n is concentrated on the unit interval $(0,1)$.

4.3 Edge Trimming

Secondly, one can apply *edge trimming* to G in order to obtain G^t, i.e. one deletes edges in G by the procedure given as follows:

1. Choose a threshold $\alpha \geq k$ and define $\mathscr{H} = \{u \in G \mid \deg_G(u) > \alpha\}$
2. For every u in $\mathscr{H}$, let $v_1^u, \ldots, v_{\deg_G(u)}^u$ be a reordering of $N_G(u)$ such that $\deg_G(v_1^u) \geq \cdots \geq \deg_G(v_{\deg_G(u)}^u)$
3. For every u in $\mathscr{H}$ remove the edges $\{u, v_1^u\}, \ldots, \{u, v_l^u\}$ from G (if possible) where $l = \deg_G(u) - \lfloor k \log_k(\deg_G(u)) \rfloor$

Figures 1 and 2 illustrate the usefulness of trimming for the regularization of kNN graphs using the UCI glass data set (Dua and Graff, 2017).

4.4 Computational Efficiency

Jebara et al. (2009) and Ozaki et al. (2011) reported that so-called b-matching graphs, a special case of b-regular graphs, achieve higher classification accuracy than kNN graphs. However, constructing the b-matching graph takes $O(bn^3)$ time (Huang and Jebara, 2007) which is too long to be useful in practice when having large amounts of data. Therefore, regularizing G within a reasonable amount of time is desirable.

Fredman and Tarjan (1987) showed that the complexity of building G is $O(n^2 + kn \log n)$. As the number of edges in G is bounded by kn, the construction time of G^ε, G^n or G^t given G is $O(kn)$. Hence the overall construction time is dominated by the term $O(n^2 + kn \log n)$.

Note that approximate kNN graphs can be constructed in $O(kn)$ time (Beygelzimer et al., 2006; Chen et al., 2009; Ram et al., 2009; Tabei et al., 2010). Combining these with the modifications discussed above yields a graph construction algorithm having time complexity $O(kn)$.

	NN	VV	AD	Other	Total
polar	4028	1810	3621	102	9561
neutral	642	253	254	61	1210

Table 1: Absolute frequencies of POS-tags among the labeled word vectors.

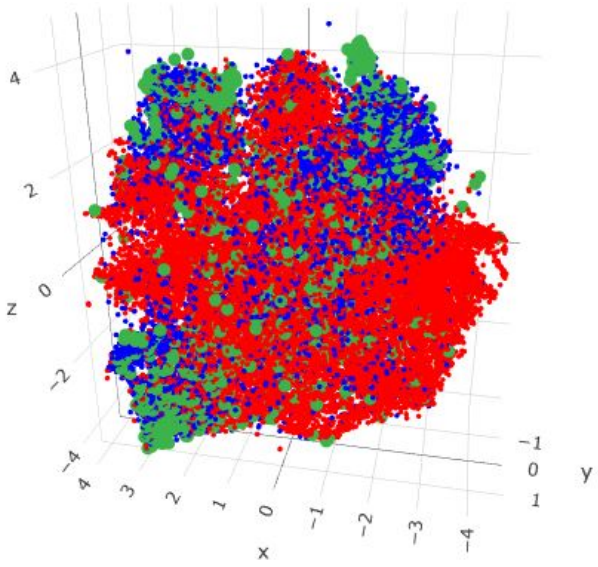

Figure 3: T-SNE plot (perplexity = 40) of the neutral (green), polar (blue) and unlabeled (red) word vectors. For sake of clarity, only 20000 red dots are shown.

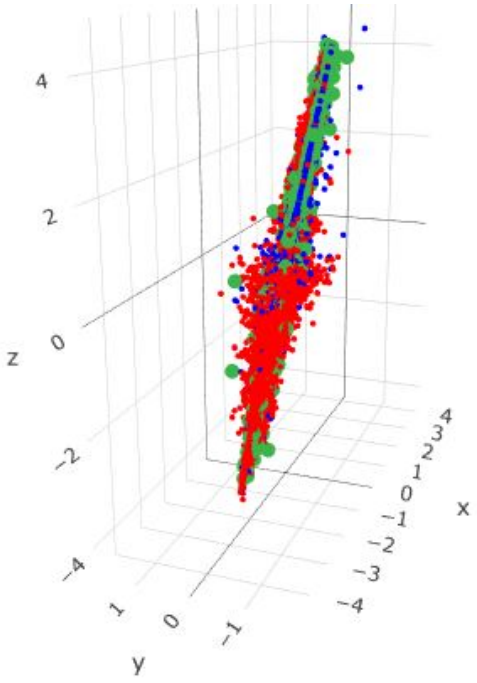

Figure 4: Figure 3 rotated around the x-axis.

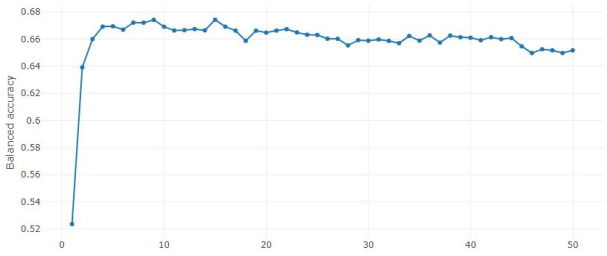

Figure 5: Balanced accuracy for label propagation on the kNN graph.

5 Experiments

In this paper, we use the lemmatized sentiment lexica introduced in (Waltinger, 2010a) and (Waltinger, 2010b) and label propagation for word-level polarity lexicon expansion.

We compare the label propagation algorithm given above on various graphs in a sentiment polarity detection task. More precisely, we consider G as in section 4 and its modifications as well as the undirected weighted kNN graph without self-loops, i.e. the graph $G_\gamma = (\mathscr{L} \cup \mathscr{U}, W_{\max})$ with similarity function $\sigma_{k,\gamma}$ using the modified version of kNN. The convergence criterion is given by $\left\|Y^{(t+1)} - Y^{(t)}\right\|_1 < 0.001$.

5.1 Dataset and Resources

As there was no public FastText model (Bojanowski et al., 2016; Joulin et al., 2016) trained on a proprietary German news corpus, we trained our own model. The resulting vocabulary size was 196972 word vectors of dimension 60. The reason for choosing FastText was the ability of the trained model to deal with out of vocabulary (OOV) words, as it is using subword character information.

The labeled word vectors are given by the lemmatized dictionaries used in (Waltinger, 2010a; Waltinger, 2010b). We assign the label 1 to the words annotated positive or negative, i.e. polar, and 0 to the words annotated neutral, where we removed the digits and the punctuation symbols from the neutral dictionary.

We prefer this lexicon over SentiWS (Remus et al., 2010) and PolArt (Klenner et al., 2009) as it is the largest one - 10771 words compared to approximately 3450 and 9380, respectively. Furthermore, SentiWS measures sentiment using the full interval $[-1,1]$, i.e. first, one has to categorise the sentiment value before one can apply label propagation.

In particular, polarity is sparsely embedded in language, i.e. a model accurately determining polarity can be used to extend sentiment dictionaries.

We choose to learn neutral vs. polar as the usually treated three-way case is significantly harder on word-level. For instance, the sentiment of 'rise' is polar, but the precise value depends heavily on the context (e.g. compare *wealth is rising* and *poverty is rising*).

5.2 Description of Dataset

The ability to embed OOV words is an integral part of our method as the labeled words are not necessarily contained in the corpus mentioned above. Figures 3 and 4 show a three-dimensional t-SNE plot (Maaten and Hinton, 2008; Van Der Maaten, 2014) of the word vectors, indicating that the dataset is lying on a low-dimensional manifold.

In total, we have 9561 data points with label 1 and 1210 data points with label 0. Only 163 ($\approx 1.5\%$) of these words have a Part-Of-Speech-tag (POS-tag) that is not noun (NN), verb (VV) or adjective (AD) (see Table 1). Therefore we only consider unlabeled words whose POS-tag is one of these three, reducing the amount of unlabeled points to 85759.

We randomly draw a test set of 3000 words from the set of unlabeled points. The test set is labeled by one of the authors. 362 words ($\approx 12.1\%$) were assigned "polar".

Comparing with an independent annotator, we have an inter-annotator agreement of Cohen's $\kappa = 0.4682$ showing that word-level sentiment analysis is a very hard to perform task, even for humans. Consequently, one cannot expect a model predicting sentiment to be performing as well as prediction models in different areas of Machine Learning.

This is probably due to the fact that sentiment is subjective and thus influenced by the emotional association of words to experiences of the individual annotator. There are even studies that suggests that the voice and audio signal is as important as the text for semantic purposes. A more fundamental fact is that sentiment in human language is better identified given the context, thus rendering the analysis of word-level sentiment even harder.

5.3 Dimensionality and Information Content of the Embedded Data

Given the ever increasing dimensionality of embeddings, from about 300 in the early Word2Vec models to more than 3000 in the most recent contextualized embeddings like ELMo (Peters et al.,

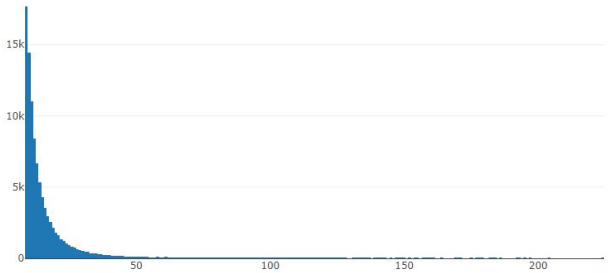

Figure 6: Distribution of degree in G.

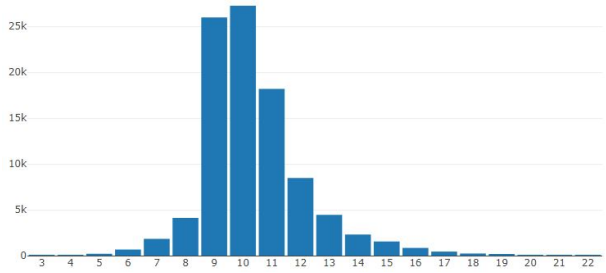

Figure 7: Distribution of degree in G^t, $\alpha = 13$.

2018), we study the cumulative explained variation of the word embeddings given by Principal Component Analysis (PCA) (Pearson, 1901) and examine it for decreasing dimension of the target space.

In every case, there is a decay starting out slowly, followed by a very sharp drop suggesting that most of the critical information content of the given word embedding is lying on a low-dimensional manifold.

5.4 Class Balancing and Parameters

Instead of transforming our labels to -1 and 1 (recall section 3.3), we normalized the labels by class size, i.e. we used $-1/1210$ for the neutral

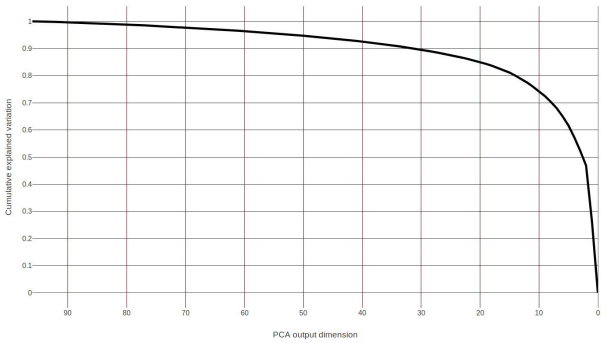

Figure 8: Cumulative explained variation of PCA on our data embedded using GloVe.

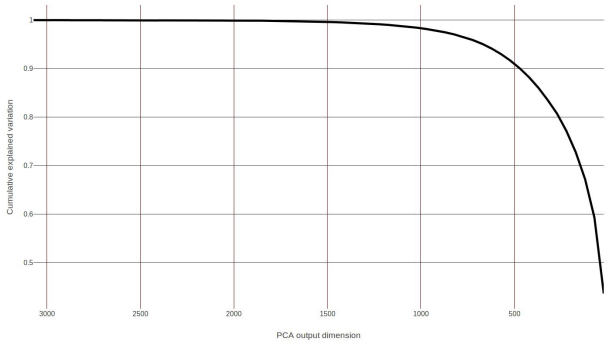

Figure 9: Cumulative explained variation of PCA on our data embedded using ELMo.

words and $1/9561$ for the polar words.

For all experiments, we use the same 9NN graph G as $k = 9$ maximizes the balanced accuracy (see Figure 5). Given the near linear time complexity of modifying G, we obtained an optimal parameter configuration using binary search.

Note that word vector models trained on a very large vocabulary, OOV words almost never occur. Hence, we also compare our self-trained embedding with different pre-trained ones.

5.5 Comparison of Word Vector Embeddings and Classification Results

Training a word vector model on the corpus at hand is usually an expensive and rewarding step at the same time. We compare our model with three pretrained word embeddings:

- FastText,
- ELMo and
- GloVe (Pennington et al., 2014).

Note that ELMo is a contextualized representation model embedding a word within its sentence. As we are working on word-level, each sentence is the word itself. The results are shown in table 2.

Despite being the lowest-dimensional, our self-trained model captures the nuances of our corpus better than the other pretrained models. Further, we can see that ELMo, one of the most recent contextualized word embedding models, clearly outperforms FastText and GloVe, whereas the latter two roughly score the same.

Table 3 shows the result for G_γ, G and its modifications using the parameters maximizing the F1 score. We can see that G_γ is performing worse than G and its modifications. In particular, removing hubs via edge normalization or trimming is improving classification performance.

Embedding	Dimension	F1	Recall	Precision	Bal. Acc.
FastText (self-trained)	60	**0.3405**	0.6381	**0.2322**	**0.6743**
GloVe	96	0.2084	0.4199	0.1386	0.5308
FastText (pre-trained)	300	0.2016	0.2376	0.1752	0.5420
ELMo	3072	0.2602	**0.6492**	0.1627	0.5954

Table 2: F1 score and balanced accuracy for G with different word embeddings to transform our data into high-dimensional vectors.

Underlying Graph	F1	Recall	Precision	Bal. Acc.
G_γ, $\gamma = 14$	0.3317	0.6630	0.2212	0.6713
G	0.3405	0.6381	0.2322	0.6743
G^ε, $\varepsilon = 110$	0.3410	0.6381	0.2326	0.6746
G^n	0.3437	0.6575	0.2326	0.6799
$(G^\varepsilon)^n$, $\varepsilon = 110$	**0.3449**	0.6602	**0.2334**	0.6813
G^t, $\alpha = 13$	0.3428	0.6685	0.2305	0.6811
$(G^\varepsilon)^t$, $\varepsilon = 120, \alpha = 12$	0.3437	**0.6740**	0.2306	**0.6827**

Table 3: F1 score and balanced accuracy for G_γ, G and G with different combinations of the modifications discussed in section 4. $(G^\varepsilon)^n$ indicates that edge normalization was applied after ε-sparsification.

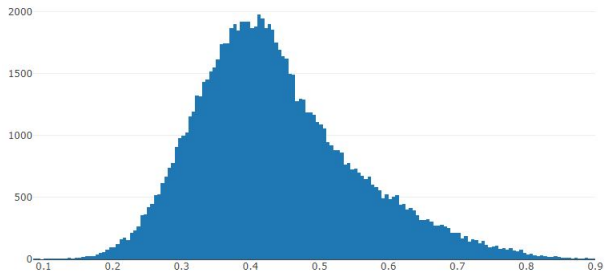

Figure 10: Distribution of weighted degree in G^n.

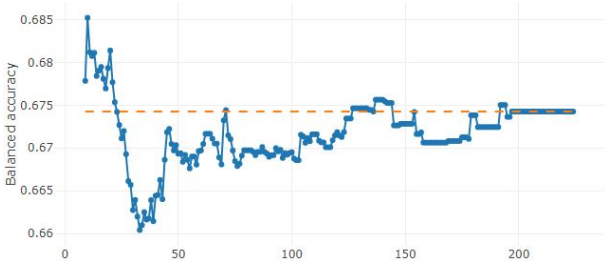

Figure 11: Balanced accuracy for G^t, $9 \leq \alpha \leq 224$. The dotted line shows the balanced accuracy for G.

5.6 Improvement of Graph Construction

In Figure 6 we can see that G not only contains vertices of degree 9, but also of degree 20 times as large. After trimming the edges, the graph is close to a 9- or 10-regular graph (see Figure 7). In particular, the maximum degree is 22, a little more than twice the most frequent degree arising in G^t.

Figure 10 shows the distribution of the weighted degree in G^n, the normalized version of G. Again, the maximum degree is a little more than twice the most frequent degree arising, whereas the minimum degree is comparatively small, i.e. the graph is not close to a regular weighted graph. However, the shape of the distribution is quite similar to the shape seen in Figure 7.

Figure 11 shows the balanced accuracy for G^t where α is ranging from 9, the minimum degree in G, to 224, the maximum degree. For small α, G^t is close to a regular graph, i.e. hubs were successfully eliminated yielding a good result. Furthermore, for large α, G^t is very similar to G and hence the result is approximately the same. However, there is a notable global minimum around $\alpha = 35$, suggesting that hub removal should be done either completely or not at all.

5.7 Towards the Fully Connected Graph

Due to the high amount of memory needed, we cannot construct the fully connected weighted graph proposed by Zhu and Ghahramani (2002), that is the graph given by taking the similarity matrix W along with the similarity function σ_γ as adjacency

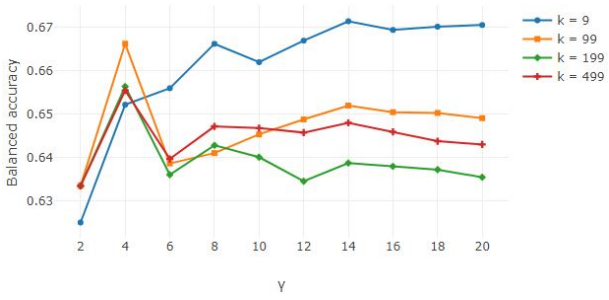

Figure 12: Bal. Accuracy for G_γ, $1 \leq \frac{\gamma}{2} \leq 10$, and multiple values for k.

matrix A. However, the weighted kNN graph G_γ with large k is a good approximation as $\sigma_\gamma(x_i, x_j)$ is strictly decreasing in $\|x_i - x_j\|_2$ and hence, only the edges having small weight are missing.

As an example, Figure 12 shows the balanced accuracy for $k \in \{9, 99, 199, 499\}$ ($k = 499$ is very close to the maximum value possible on our hardware). We can see that large k harms the classification, thus confirming the results on sparse adjacency matrices mentioned in section 3.2.

We do not rule out the fact that there could be a state change as $k \approx n$ where the information flow improves drastically and causes the SSL classification performance to spike. We leave this as an open question for future work.

6 Conclusion

In this paper, we study label propagation for sentiment detection on word vectors obtained by training a FastText model as well as by using pre-trained models, which clearly perform worse. We showed empirically that the unweighted 9NN graph performs better on the given task than its weighted counterpart and the approximation of the fully connected weighted graph.

Furthermore, we propose improvements to state-of-the-art methods for the construction of the underlying graph. and show that properly chosen anti-hub routines and mild ε-sparsification improves the result. In particular, edge trimming is a fast algorithm to transform a kNN graph into a more regular one.

7 Future Work

Possible directions for future research include the development of an online label propagation algorithm based on entropy and data quantization (in the spirit of (Valko et al., 2012)). The goal is to improve classification performance for situations where the word vector embedding of the given data does not fulfill the cluster assumption perfectly. Furthermore, the ability of being able to deal with streaming data is a highly attractive add-on for practical applications of SSL models.

Another interesting idea is the search for metrics quantifying the cluster assumption for the embedded data, as discussed above. This can be supplemented by an analysis of the performance of label propagation conditioned on the scores provided by the metrics found above and hence, by the relevance of the word embedding.

Datasets which can be used to examine the performance of the given SSL algorithm include the annotations on polarity shifters by (Schulder et al., 2018) and the domain-specific corpora for computational social science by (Hamilton et al., 2016).

References

[Bengio et al.2006] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 2006. 11 label propagation and quadratic criterion.

[Beygelzimer et al.2006] Alina Beygelzimer, Sham Kakade, and John Langford. 2006. Cover trees for nearest neighbor. In *Proceedings of the 23rd international conference on Machine learning*, pages 97–104. ACM.

[Bojanowski et al.2016] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

[Chapelle et al.2009] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

[Chen et al.2009] Jie Chen, Haw-ren Fang, and Yousef Saad. 2009. Fast approximate knn graph construction for high dimensional data via recursive lanczos bisection. *Journal of Machine Learning Research*, 10(Sep):1989–2012.

[de Sousa et al.2013] Celso André R de Sousa, Solange O Rezende, and Gustavo EAPA Batista. 2013. Influence of graph construction on semi-supervised learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 160–175. Springer.

[dos Santos and Gatti2014] Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

[Dua and Graff2017] Dheeru Dua and Casey Graff. 2017. UCI machine learning repository.

[Fredman and Tarjan1987] Michael L Fredman and Robert Endre Tarjan. 1987. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615.

[Gamon2004] Michael Gamon. 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics*, page 841. Association for Computational Linguistics.

[Giulianelli2017] Mario Giulianelli. 2017. Semi-supervised emotion lexicon expansion with label propagation and specialized word embeddings. *CoRR*, abs/1708.03910.

[Goldberg and Zhu2006] Andrew B Goldberg and Xiaojin Zhu. 2006. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 45–52. Association for Computational Linguistics.

[Hamilton et al.2016] William L Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 595. NIH Public Access.

[Huang and Jebara2007] Bert Huang and Tony Jebara. 2007. Loopy belief propagation for bipartite maximum weight b-matching. In *Artificial Intelligence and Statistics*, pages 195–202.

[Jebara et al.2009] Tony Jebara, Jun Wang, and Shih-Fu Chang. 2009. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 441–448. ACM.

[Joulin et al.2016] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

[Klenner et al.2009] Manfred Klenner, Angela Fahrni, and Stefanos Petrakis. 2009. Polart: A robust tool for sentiment analysis. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 235–238.

[Maaten and Hinton2008] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

[Matsumoto et al.2005] Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. 2005. Sentiment classification using word sub-sequences and dependency sub-trees. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 301–311. Springer.

[Mikolov et al.2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Ozaki et al.2011] Kohei Ozaki, Masashi Shimbo, Mamoru Komachi, and Yuji Matsumoto. 2011. Using the mutual k-nearest neighbor graphs for semi-supervised classification on natural language data. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 154–162. Association for Computational Linguistics.

[Pang et al.2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

[Pearson1901] Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[Peters et al.2018] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

[Radovanović et al.2010] Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep):2487–2531.

[Ram et al.2009] Parikshit Ram, Dongryeol Lee, William March, and Alexander G Gray. 2009. Linear-time algorithms for pairwise statistical problems. In *Advances in Neural Information Processing Systems*, pages 1527–1535.

[Rao and Ravichandran2009] Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 675–682. Association for Computational Linguistics.

[Ravi and Diao2016] Sujith Ravi and Qiming Diao. 2016. Large scale distributed semi-supervised learning using streaming approximation. In *Artificial Intelligence and Statistics*, pages 519–528.

[Remus et al.2010] Robert Remus, Uwe Quasthoff, and Gerhard Heyer. 2010. Sentiws-a publicly available german-language resource for sentiment analysis. In *LREC*. Citeseer.

[Ren et al.2012] Yong Ren, Nobuhiro Kaji, Naoki Yoshinaga, Masashi TOYODA, and Masaru KITSUREGAWA. 2012. Semi-supervised sentiment classification in resource-scarce language: A comparative study (). . *DE,* , 112(172):59–64.

[Schulder et al.2018] Marc Schulder, Michael Wiegand, Josef Ruppenhofer, and Stephanie Köser. 2018. Introducing a lexicon of verbal polarity shifters for english.

[Tabei et al.2010] Yasuo Tabei, Takeaki Uno, Masashi Sugiyama, and Koji Tsuda. 2010. Single versus multiple sorting in all pairs similarity search. In Masashi Sugiyama and Qiang Yang, editors, *Proceedings of 2nd Asian Conference on Machine Learning*, volume 13 of *Proceedings of Machine Learning Research*, pages 145–160, Tokyo, Japan, 08–10 Nov. PMLR.

[Taboada et al.2011] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

[Valko et al.2012] Michal Valko, Branislav Kveton, Ling Huang, and Daniel Ting. 2012. Online semi-supervised learning on quantized graphs. *arXiv preprint arXiv:1203.3522*.

[Van Der Maaten2014] Laurens Van Der Maaten. 2014. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245.

[Vega-Oliveros et al.2014] Didier A Vega-Oliveros, Lilian Berton, Andre Mantini Eberle, Alneu de Andrade Lopes, and Liang Zhao. 2014. Regular graph construction for semi-supervised learning. In *Journal of physics: Conference series*, volume 490, page 012022. IOP Publishing.

[Waltinger2010a] Ulli Waltinger. 2010a. Germanpolarityclues: A lexical resource for german sentiment analysis. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta, May. electronic proceedings.

[Waltinger2010b] Ulli Waltinger. 2010b. Sentiment analysis reloaded: A comparative study on sentiment polarity identification combining machine learning and subjectivity features. In *Proceedings of the 6th International Conference on Web Information Systems and Technologies (WEBIST '10)*, Valencia, Spain, April.

[Zhu and Ghahramani2002] Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation.

[Zhu2008] X Zhu. 2008. Semi-supervised learning literature survey, department of computer sciences, university of wisconsin. Technical report, Madison, 2008 (Technical Report, 1530).