Learning Word Embeddings from Tagging Data: A methodological comparison

Thomas Niebler[1], Luzian Hahn[1], and Andreas Hotho[1,2]

[1] Data Mining and Information Retrieval Group, University of Würzburg (Germany)
{niebler, hotho}@informatik.uni-wuerzburg.de
luzian.hahn@stud-mail.uni-wuerzburg.de
[2] L3S Research Center Hanover (Germany)

Abstract The semantics hidden in natural language are an essential building block for a common language understanding needed in areas like NLP or the Semantic Web. Such information is hidden for example in lightweight knowledge representations such as tagging systems and folksonomies. While extracting relatedness from tagging systems shows promising results, the extracted information is often encoded in high dimensional vector representations, which makes relatedness learning or word sense discovery computationally infeasible. In the last few years, methods producing low-dimensional vector representations, so-called word embeddings, have been shown to yield extraordinary structural and semantic features and have been used in many settings. Up to this point, there has been no in-depth exploration of the applicability of word embedding algorithms on tagging data. In this work, we explore different embedding algorithms with regard to their applicability on tagging data and the semantic quality of the produced word embeddings. For this, we use data from three different tagging systems and evaluate the vector representations on several human intuition datasets. To the best of our knowledge, we are the first to generate embeddings from tagging data. Our results encourage the use of word embeddings based on tagging data, as they capture semantic relations between tags better than high-dimensional representations and make learning with tag representations feasible.

1 Introduction

Automatically assessing the degree of semantic relatedness between words, i.e., the relatedness of their actual meanings, in such a way that it fits human intuition is an important task with a variety of applications, such as ontology learning for the Semantic Web [3], tag recommendation [15] or semantic search [13]. Semantic relatedness information of words has been extracted from a variety of sources like plain text [7], website navigation [21, 27] or social metadata [5, 8, 17]. Among others, tagging data from social tagging systems like BibSonomy[3] or Delicious[4] are useful to extract high-quality semantic relatedness information, e.g., for ontology learning [3].

Traditionally, assessing the degree of semantic relatedness between tags utilizes sparse, high-dimensional vector representations of those tags, which are constructed

[3] https://www.bibsonomy.org
[4] http://www.delicious.com

In: M. Leyer (Ed.): Proceedings of the LWDA 2017 Workshops: KDML, FGWM, IR, and FGDB.
Rostock, Germany, 11.-13. September 2017, published at http://ceur-ws.org

from tag contexts based on posts in social tagging systems [8]. The semantic relatedness can then be estimated using the cosine measure of the corresponding tag vectors [17]. Finally, evaluating the quality of the estimated scores is usually performed by directly correlating them to human intuition [6, 11, 25]. In recent years, many techniques have been proposed to represent words by dense, low-dimensional vectors [20, 23, 28]. These so-called *word embeddings* have been shown to yield extraordinary structural features [16, 19] and are applied in machine translation or text classification. Furthermore, word embeddings often outperform high-dimensional representations in tasks such as measuring semantic relatedness [1, 16].

Problem Setting. Traditionally, tags are represented by sparse, high-dimensional vectors [8, 26]. However, although Cattuto et al. have shown that tagging data contain meaningful semantics [8], the correlation of semantic relatedness scores from those vectors with human intuition still leaves room for improvement. Furthermore, the high dimensionality of those representations renders many algorithms using them computationally expensive.[5] Up to this point, there have been no extensive attempts to generate word embeddings from social tagging data. All prior studies rely on high dimensional tagging vectors or reduce the vector space arbitrarily by cutting the dimensionality of the space by a fixed number, which in turn decreases the fit of the resulting relatedness scores to human intuition.

Contribution. We contribute a thorough exploration of the applicability and optimization of three well-known embedding algorithms on tagging data. We first analyze the parameters of each algorithm, before we optimize these settings to produce the best possible word embeddings from tagging data. Then, we compare the embeddings of each algorithm with each other as well as with traditional sparse representations by evaluating them on human intuition. We show that all produced embeddings outperform high-dimensional vector representations. We discuss the results in the light of other semantic relatedness approaches and show that we reach competitive results, on par with recent work on extracting semantic relatedness, which opens another high quality source of information for semantic relatedness.

Structure of this work. We first cover the related work in Section 2 and any essential theoretical background of this work in Section 3. Afterwards, we investigate three well-known algorithms with respect to their applicability on tagging data in Section 4. In Section 5, we describe the datasets we used in our experiments. Section 6 outlines our experiments, where we compare all generated vector representations with regard to their semantic content. Section 8 concludes this work.

2 Related Work

The related work to this paper can be roughly put in two groups: Word Embedding algorithms and semantics of tagging data as well as their applications.

Word Embeddings. The concept of *word embeddings*, i.e., word representations in low dimensional vector spaces dates back at least to 1990, when Deerwester presented LSA [9], which by factorizing a term-document matrix effectively produced a dimension reduction of the term vector space. In 2003, Bengio et al. presented their *neural probabilistic language model* [2]. The goal of this work was to overcome

[5] This is also sometimes referred to as the *curse of dimensionality*

the *curse of dimensionality* and learn distributed word representation in a low-dimensional vector space. However, the wide-spread use of word embeddings only really took off in 2013, when Mikolov et al. presented a similar, yet scalable and fast approach to learn word embeddings [20]. Generally, such methods train a model to predict a word from a given context [2, 20]. Other embedding methods focus on factorizing a term-document matrix [9, 23]. In [1], Baroni et al. showed that all those methods generally exhibit a notably higher correlation with human intuition than the standard high-dimensional vector representations proposed in [26]. There also exist several *graph embedding algorithms*. The LINE algorithm [28] attempts to preserve the first- and second-order proximity of nodes in their corresponding embedding relations. Perozzi et al. [24] proposed "DeepWalk", an approach based on random walks on graphs and the subsequent embedding using Word2Vec.

Social Tagging Systems. In [12], Golder and Huberman noted that with increasing use, usage data from *social tagging systems* exhibited an emerging structure, which was later called a *folksonomy* [29]. Mika noted that these emerging structures, i.e., folksonomies, could even represent *light-weight ontologies* [18]. Using the folksonomy structure, it was possible to extract information about semantic relatedness between tags [8, 17]. The evaluation of this semantic relatedness information on human intuition showed that tagging data contain a considerable amount of semantic information, thus enabling further applications of tagging data. Applications of these emerging structures can be found in tag recommendation [15], ontology learning [3] and tag sense discovery algorithms [22].

3 Technical Background

In the following, we will describe the technical background for this paper. First, we define the term folksonomy. Secondly, we introduce how to extract information about semantic relatedness from folksonomies.

Folksonomies. Folksonomies are the data structures emerging from social tagging systems. The term has been coined by Van der Wal in 2005 as a portmanteu of "folks" and "taxonomy" [29]. In these systems, users collect resources and annotate them with freely chosen keywords, so-called *tags*. Examples are BibSonomy, Delicious, FlickR or last.fm. We follow the definition given by [14]:

A ***folksonomy*** *is a tuple* (U, T, R, Y) *of sets* U, T, R *and a tripartite relation* $Y \subseteq U \times T \times R$. *The sets* U, T *and* R *represent the sets of users, tags and resources, respectively, while* Y *represents the set of tag assignments. A* post *is the collection of tag assignments with the same user and same resource.*

Extracting Semantic Relatedness from Folksonomies. After Golder and Huberman argued that the emerging structure of folksonomies contains considerable semantic information [12], Cattuto et al. proposed a way to extract this information [8]. They used a context-co-occurrence based vector representation for the tags and experimented with different context choices, such as tag-tag-context or tag-resource-context, i.e., all assigned tags of a posted resource by either a specific user or all users. Both of these context choices were shown to estimate human-perceived semantic relatedness better than other context variants. In this work, we generally use the tag-tag-context. The resulting vector representations follow the definition

given in [26] and are based on the co-occurrence counts of tags in their respective contexts. More concretely, a vector representation v_i of a tag $t_i \in V$ in a given vocabulary is then a $|V|$-dimensional vector, where $v_{ij} := \#coocc_{post}(i, j)$. To finally receive a notion of the degree of semantic relatedness between two tags i and j, one can compare the corresponding vectors v_i and v_j using the cosine measure $cossim(v_i, v_j) := \frac{\langle v_i, v_j \rangle}{\|v_i\| \cdot \|v_j\|}$ [17].

4 Applicability of Embedding Algorithms on Tagging Data

This section describes the different embedding algorithms that we explored. For each algorithm, we give a short summary, enumerate the parameters for each model and shortly discuss how it can be applied to tagging data.

Word2Vec The most well-known embedding algorithm used in this work is the Word2Vec algorithm [20], which is actually comprised of two algorithms, SkipGram and CBOW (Cumulative Bag of Words).[6] Word2Vec trains a shallow neural network on word sequences to predict a word from its neigbors in a given context window.
Parameterization. Word2Vec takes two parameters. The first parameter is the *window size*, which determines the amount of neighboring words in a sequence considered as context from which a word will be predicted. The second parameter is the number of *negative samples* per step. This is done to decrease complexity of solving the proposed model by employing a noise contrastive estimation approach.
Applicability. The Word2Vec algorithm normally processes sequential data. However, the sequence of tags normally does not hold any meaning, so this could possibly pose a problem if the window size is chosen too small. In order to be able to apply Word2Vec on tagging data, we grouped the tag assignments into posts and fed the random succession of tags as sentences into the algorithm.

GloVe GloVe is an unsupervised learning algorithm for obtaining vector representations for words [23]. Its main objective was to capture semantic relations such as $king - man + woman \approx queen$. Training is performed on aggregated global word-word co-occurrence statistics from a corpus.
Parameterization. The main parameters of the GloVe algorithm are x_{max} and α. x_{max} denotes an influence cutoff for frequent tags while α determines the importance of infrequent tags. According to [23], GloVe worked best for $x_{max} = 100$ and $\alpha = 0.75$. We will choose these as initial values in our experiments.
Applicability. Since GloVe depends on co-occurrence counts of words in a corpus, it is very easy to apply on tagging data. For this, we construct the tag-tag-context co-occurrence matrix and can then directly feed it into the algorithm.

LINE The goal of the LINE embedding algorithm is to create graph embeddings where the first- and second-order proximity of nodes are preserved [28]. The *first-order proximity* in a network is the *local* pairwise proximity of two nodes, i.e., the

[6] In the course of this work, every time we refer to Word2Vec, we talk about the CBOW algorithm, as is recommended by [20] for bigger datasets.

weight of an edge connecting these two nodes. The *second-order proximity* of two nodes in a network is the similarity between their *first-order neighborhoods*.
Parameterization. LINE takes two different parameters: The amount of edge *samples per step* and the amount of *negative samples* per edge. To decrease complexity of solving the proposed model in [28], the authors employed a noise contrastive estimation approach as proposed by [20] using negative sampling. Furthermore, to avoid high edge weights to outweigh lower weights by letting the gradient explode or vanish, LINE employs a sampling process of edges and then ignores their weights instead of actually using the edge weights in its objective function.
Applicability. Similar to GloVe, this algorithm processes a network with weighted edges, such as a co-occurrence network. Thus, we only have to construct the co-occurrence network from the tagging data and apply LINE on that network.

Common Parameters While each of the mentioned algorithms can be tuned with a set of different parameters, they have some parameters in common. First, the *embedding dimension* determines the size of the produced vectors. A higher embedding dimension allows for more degrees of freedom in the expressiveness of the vector, i.e., it can encode more information about word relations. Standard ranges for embedding dimensions are between 25 and 300. Secondly, the *initial learning rate* of an algorithm determines its convergence speed. Fine-tuning that parameter is crucial to receive optimal results, because if chosen badly, the learning process either converges very slowly or might be unable to converge at all.

5 Datasets

In this work we use two different kinds of datasets to evaluate embedding algorithms on tagging data. That is, the actual *tagging datasets* which provide tagging metadata and *human intuition datasets* (HIDs) which we employ to evaluate semantic relatedness. In the following we first describe three datasets containing tagging data from which we later derive tag embeddings. Then we introduce all human intuition datasets containing human-assigned scores of similarities to word pairs.

5.1 Tagging Datasets to Derive Word Embeddings

We study datasets of three public social tagging systems. In order to ensure a minimum level of commonly accepted meaning of all tags, each dataset is restricted to the top 10k tags. Additionally, we only considered tags from users who have tagged at least 5 resources and resources which have been used at least 10 times. We also removed all invalid tags, e.g., containing whitespaces or unreadable symbols.
BibSonomy. The social tagging system BibSonomy provides users with the possibility to collect bookmarks (links to websites) or references to scientific publications and annotate them with tags [4]. We use a freely available dump of BibSonomy, covering all tagging data from 2006 till the end of 2015.[7] After filtering, it contains 9,302 distinct tags, assigned by 3,270 users to 49,654 resources in 630,962 assignments.

[7] `http://www.kde.cs.uni-kassel.de/bibsonomy/dumps/`

Delicious. Like BibSonomy, Delicious is a social tagging system, where users can share their bookmarks and annotate them with tags. We use a freely available dataset from 2011 [30].[8] Delicious has been one of the biggest adopters of the tagging paradigm and due to its audience, contains tags about design and technical topics. After filtering, the Delicious dataset contains 10,000 tags, which were assigned by 1,685,506 users to 11,486,080 resources in 626,690,002 assignments.
CiteULike. We took a snapshot of the official CiteULike page from September 2016.[9] Since CiteULike describes itself as a "free service for managing and discovering scholarly references", it contains tags mostly centered around research topics. After filtering, the CiteULike dataset contains 10,000 tags, which were assigned by 141,395 users to 4,548,376 resources in 15,988,259 assignments.

5.2 Human Intuition Datasets (HIDs)

As a gold standard for semantic relatedness as it is perceived by humans, we use several datasets with human-generated relatedness scores for word pairs. In the following, we will describe all of the used datasets briefly.
WS-353. The WordSimilarity-353[10] dataset consists of 353 pairs of English words and names [10]. Each pair was assigned a relatedness value between 0.0 (no relation) and 10.0 (identical meaning) by 16 raters, denoting the assumed common sense semantic relatedness between two words. Finally, the total rating per pair was calculated as the mean of each of the 16 users' ratings. This way, WS-353 provides a valuable evaluation base for comparing our concept relatedness scores to an established human generated and validated collection of word pairs.
MEN. The MEN Test Collection [6] contains 3,000 word pairs together with human-assigned similarity judgments, obtained by crowdsourcing using Amazon Mechanical Turk[11]. Contrary to WS-353, the similarity judgments are relative rather than absolute. Raters were given two pairs of words at a time and were asked to choose the pair of words was more similar. The score of the chosen pair, i.e., the pair of words that was more similar, was then increased by one. Each pair was rated 50 times, which leads to a score between 0 and 50 for each pair.
Bib100. The Bib100 dataset has been created in order to provide a more fitting vocabulary for the research and computer science oriented tagging data that we investigate.[12] It consists of 122 words in 100 pairs, which were judged 26 times each for semantic relatedness using scores from 0 (no similarity) to 10 (full similarity).
MTurk. In [25], Radinsky et al. created an evaluation dataset specifically for news texts.[13] We use this dataset as a topically remote evaluation baseline in order to get a notion how intrinsic semantic relations are captured by both the tagging data and the generated embeddings. The dataset at hand consists of 287 word pairs and 499 words. 23 humans judged relatedness on a scale from 1 (unrelated) to 5 (related).

[8] http://www.zubiaga.org/datasets/socialbm0311/
[9] http://www.citeulike.org/faq/data.adp
[10] http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/wordsim353.html
[11] http://clic.cimec.unitn.it/~elia.bruni/MEN
[12] http://www.dmir.org/datasets/bib100
[13] http://www.kiraradinsky.com/Datasets.html

6 Experimental Setup and Results

In the following, we describe the conducted experiments and present the results for each experiment. Due to space limitations, we only report results for MEN. [14]

6.1 Preliminaries

Evaluating Word Vector Representations. Very often, the quality of semantic relatedness encoded in word vectors is assessed by how well it fits human intuition. Human intuition is collected in HIDs as introduced in Section 5. The most widely-used method to evaluate semantic relatedness on such datasets is to compare human scores of the semantic relatedness between two words with the cosine similarity scores of the corresponding word vectors. The comparison is done by calculating the Spearman rank correlation coefficient ρ, which compares two ranked lists of word pairs induced by the human relatedness scores and the cosine scores [1, 23].
Baseline: Tag-Tag-Context Co-Occurrence Vectors. As a baseline, we produced high dimensional co-occurrence counting vectors from all three tagging datasets. As described in Section 3, co-occurrence of tags was counted in a tag-tag-context, i.e., the context of a tag was given as the other tags annotated to a given resource by a certain user [8]. Since there is no option to vary the dimension of the tag-tag-context co-occurrence vectors except truncating the vocabulary, we only report the values for a truncated vocabulary of 10,000 tags in Table 1. Still, we give all of the reported results as baselines in the subsequent figures.
Parameter Settings. For each of the following algorithms, we conducted the experiments as follows: As initial parameter setting, we used the standard settings that come with the implementation of each algorithm. The corresponding values are given in Table 2. We then varied the initial learning rate for each algorithm in the range of 0.01 to 0.1 in steps of 0.01. After that, we varied the embedding dimension on the set of $\{10, 30, 50, 80, 100, 120, 150, 200\}$. For Word2Vec and LINE, we now varied the number of negative samples on the set of $\{2, 5, 8, 12, 15, 20\}$. For GloVe, we varied $x_{max} \in \{25, 50, \ldots, 200\}$ and $\alpha \in \{0.5, 0.55, \ldots, 1\}$ simultaneously. Finally, for Word2Vec, we varied the context window size between $\{1, 3, 5, 8, 10, 13, 16, 20\}$, while for LINE, we varied the number of samples per step on $\{1, 10, 100, 1000, 10000\} \cdot 10^6$. To rule out influence of a random embedding initialization, each experiment was performed 10 times and the mean result was reported. After each experiment, we chose the best performing parameter settings on the respective tagging datasets across the four evaluation datasets and used them for all other experiments.

6.2 Embedding Evaluation Results

We will now present the evaluation results. For each algorithm, Table 3 gives the parameter settings which produced the highest-scoring embeddings. In each figure, we report both the evaluation results of the embeddings for a given parameter as well as the corresponding baselines produced by the high-dimensional vector representations given in Table 1.

[14] All result figures are publicly available at `http://www.dmir.org/tagembeddings`.

Table 1: Spearman correlation values for the co-occurrence baseline. For all evaluation datasets, we gave the total number of pairs in the original dataset and the number of matched pairs, i.e., where both words were present in the tagging corpus.

	WS-353 (353)	MEN (3000)	MTurk (287)	Bib100 (100)
BibSonomy	0.433 (158)	0.415 (463)	0.604 (62)	0.623 (100)
Delicious	0.486 (202)	0.577 (1376)	0.508 (103)	0.632 (94)
CiteULike	0.186 (139)	0.423 (404)	0.469 (53)	0.270 (87)

Table 2: Initial parameter values for each algorithm.

Algorithm	initial learning rate	dimension	samples per step	negative samples	(x_{max}, α)	window size
LINE	0.025	100	$100 \cdot 10^6$	5	-	-
GloVe	0.05	100	-	-	(100, 0.75)	-
Word2vec	0.025	100	-	5	-	5

Table 3: Best parameter values for each algorithm for the MEN dataset on Delicious data.

Algorithm	initial learning rate	dimension	samples per step	negative samples	(x_{max}, α)	window size
LINE	0.1	100	$100 \cdot 10^6$	15	-	-
GloVe	0.1	120	-	-	(100, 0.75)	-
Word2vec	0.1	100	-	20	-	5

Word2Vec. Although Word2Vec is meant to be applied on sequential data, as opposed to the bag-of-words nature when assigning tags, the generated embeddings yielded better correlation scores with human intuition than their high-dimensional counterparts. However, we did not shuffle the tag sequence in posts, which is left to future work. Figure 1a shows that fine-tuning the *initial learning rate* exhibits a great effect on the quality of word embeddings from BibSonomy, with general peak performance at $\alpha = 0.1$, while Delicious data seem unaffected. Increasing the *embedding dimension* only improves the embeddings' semantic content up to a certain point, which is mostly reached at around a very low number of dimensions between 30 and 50 (Figure 1b). Anything above does not notably increase performance of the embeddings. The number of negative samples seems sufficiently high at 10 samples and even earlier for Delicious and CiteULike (Figure 1c). The *context size* had negligible impact on the semantic content of the generated embeddings (Figure 1d).
GloVe. GloVe generates embeddings from co-occurrence data. As mentioned in Section 4, GloVe is parameterized by the learning rate, the dimension of the generated embeddings as well as by the weighting parameters x_{max} and α, which regulate the importance of low-frequency co-occurrences in the training process. While the *learning rate* does not show a great effect on embeddings generated from Delicious data, fine-tuning influences the semantic content of CiteULike and BibSonomy embeddings notably (Figure 3a). Mostly, peak performance is reached at an *embedding dimension* of 100 or even earlier, except for Delicious (Figure 3b) Furthermore, BibSonomy is quite sensitive to poor choices of x_{max} and α, i.e., if chosen too high, performance suffers greatly (Figure 3c). Delicious and CiteULike seem unaffected by those parameters, at least in our experimental ranges (Figures 3d and 3e).
LINE. LINE generates vertex embeddings from graph data, preserving the first- and second-order proximity between vertices. Its parameters are the initial learning

rate, the embedding dimension, the number of negative samples per edge and the number of samples per training step. While influence of the *initial learning rate* is visible, it is not as great as with GloVe (cf. Figure 2a). Also, the *embedding dimension* gives similar results above 50 and only lets performance suffer if chosen too small (cf. Figure 2b). Interestingly enough, Figure 2c shows that the number of *negative samples* seems to have almost no effect on the generated embeddings across all tagging datasets. In contrast, choosing the number of *samples per step* exerts great influence of the resulting embeddings, as can be seen in Figure 2d.

7 Discussion

Across all algorithms, fine-tuning the *initial learning rate* greatly improves results for embeddings based on BibSonomy, especially with GloVe. The effect of the *embedding dimension* is much less pronounced across all three embedding algorithms. Peak evaluation performance is often reached with an embedding dimension between 50 and 100 and stays quite stable with increasing dimension. Varying the number of *negative samples* influences evaluation results of BibSonomy, but only at a very high number of 20 negative samples. In contrast, Delicious and CiteULike only show small performance changes already with 3 to 5 samples. Finally, GloVe's *weighting factors* x_{max} and α negatively influence results on BibSonomy, while barely affecting evaluation performance on Delicious and CiteULike, due to BibSonomy being our smallest tagging dataset with rarely any co-occurrences above a high x_{max}.

Generally, all investigated embedding algorithms produce high-quality embeddings from tagging data. Although [8] found that tagging data contain high-quality semantic information, the high-dimensional vector representation proposed there

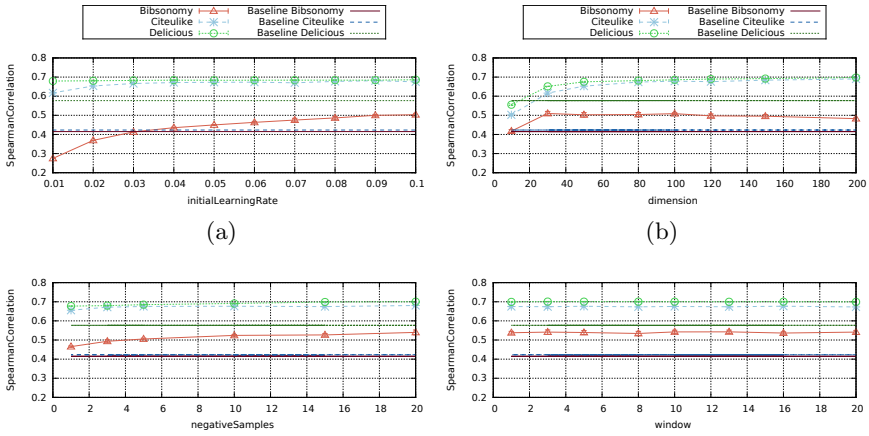


Figure 1: Evaluation results for embeddings generated by Word2Vec on MEN. For CiteULike and BibSonomy, tuning the learning rate notably improves results. The embedding dimension seems to be the best for all tagging dataset at 100; before the embedding quality of BibSonomy decreases again. As reported by [20], roughly 10 negative samples fit for small datasets such as BibSonomy, while even fewer samples fit for bigger datasets.

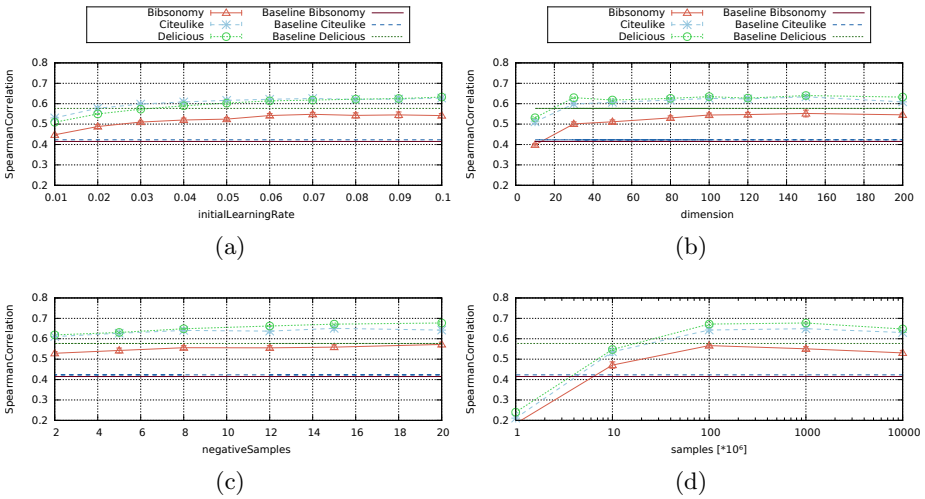
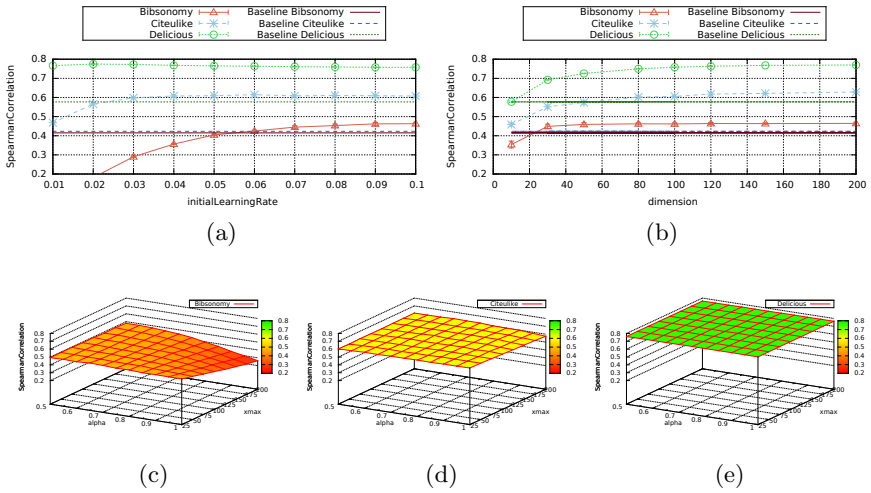


Figure 2: Evaluation results for embeddings generated by LINE on the MEN dataset. While the initial learning rate, the embedding dimension and the amount of samples per step exert a notable influence on the evaluation result, increasing the number of negative samples per edge only slightly improves results.

seems to not optimally capture this information when evaluated on human judgment (see Table 1). In contrast, the generated embeddings seem better suited to capture that information, as they mostly outperform the tag-tag-context based co-occurrence count vectors (Section 8). Furthermore, the best result achievable on WS-353 in this work is from Delicious data using the GloVe algorithm of around 0.7 (cf. Figure 4b), which is on par with with other well-known works, such as ESA [11], which is based on Wikipedia text, achieving correlation around 0.748, or the work done by Singer et al. on Wikipedia navigation [27] with the highest correlation at 0.76, but generally achieving scores around 0.71.

8 Conclusion

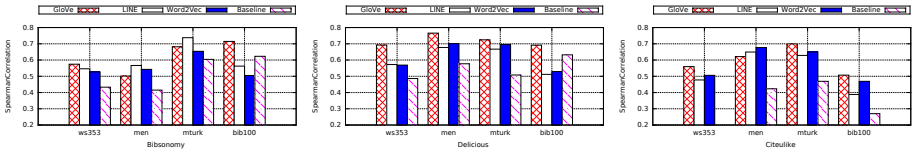
In this work, we explored embedding methods and their applicability on tagging data. We conducted parameter studies for three well-known embedding algorithms in order to achieve the best possible embeddings based on tagging data regarding their fit to human intuition of semantic relatedness. Our results indicate that i) tagging data provide a viable source to generate high-quality semantic embeddings, even on par with current state-of-the-art methods and ii) that in order to achieve competitive results, it is necessary to choose correct parameters for each algorithm instead of the standard parameters. Overall we bridged the gap between the fact that tagging data yield considerable semantic content and the current state-of-the-art methods to produce high-quality and low-dimensional word embeddings. We expect our results to be of special interest for folksonomy engineers and others working with semantics of tagging data. *Future work* includes investigation of the influence of different vector representations on tagging-based real-world applications, such as



(a) (b)

(c) (d) (e)

Figure 3: Evaluation results for embeddings generated by GloVe on the MEN dataset. The initial learning rate only influences the smaller tagging datasets, while Delicious profits most from increasing dimension. BibSonomy is influenced by a high cutoff x_{max} the most.



(a) BibSonomy (b) Delicious (c) CiteULike

Figure 4: Evaluation results for embeddings generated by the best parameter settings across the different tagging datasets. GloVe mostly produces the best embeddings and is the only algorithm to always outperform the baseline.

tag recommendations in social tagging systems, tag sense discovery and ontology learning algorithms. Furthermore, we want to try to improve the fit of tagging embeddings to human intuition by applying metric learning approaches or alignment approaches to external knowledge bases, e.g., WordNet or DBPedia.

References

[1] Marco Baroni, Gerorgiana Dinu, and German Kruszewski. "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors." In: *ACL* (2014).

[2] Yoshua Bengio et al. "A neural probabilistic language model." In: *JMLR* (2003).

[3] Dominik Benz et al. "Semantics made by you and me: Self-emerging ontologies can capture the diversity of shared knowledge." In: *WebSci*. 2010.

[4] Dominik Benz et al. "The Social Bookmark and Publication Management System BibSonomy." In: *VLDB* (2010).

[5] Kalina Bontcheva and Dominic Rout. "Making sense of social media streams through semantics: a survey." In: *Semantic Web* 5.5 (2014).
[6] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. "Multimodal Distributional Semantics." In: *JAIR* (2014).
[7] John A Bullinaria and Joseph P Levy. "Extracting semantic representations from word co-occurrence statistics: A computational study." In: *BRM* 39.3 (2007).
[8] Ciro Cattuto et al. "Semantic Grounding of Tag Relatedness in Social Bookmarking Systems." In: *ISWC*. 2008.
[9] Scott Deerwester et al. "Indexing by latent semantic analysis." In: *Journal of the American Society for Information Science* 41.6 (1990).
[10] Lev Finkelstein et al. "Placing Search in Context: the Concept Revisited." In: *WWW*. 2001.
[11] Evgeniy Gabrilovich and Shaul Markovitch. "Computing semantic relatedness using Wikipedia-based explicit semantic analysis." In: *IJCAI*. 2007.
[12] Scott Golder and Bernardo A. Huberman. "The Structure of Collaborative Tagging Systems." In: (Aug. 2005). arXiv: `cs.DL/0508082`.
[13] Mihajlo Grbovic et al. "Scalable Semantic Matching of Queries to Ads in Sponsored Search Advertising." In: *SIGIR*. 2016.
[14] Andreas Hotho et al. "Information Retrieval in Folksonomies: Search and Ranking." In: *ESWC*. Ed. by York Sure and John Domingue. 2006.
[15] Robert Jäschke et al. "Tag Recommendations in Folksonomies." In: *PKDD*. Ed. by Joost N. Kok et al. 2007.
[16] Omer Levy, Yoav Goldberg, and Israel Ramat-Gan. "Linguistic Regularities in Sparse and Explicit Word Representations." In: *CoNLL*. 2014.
[17] Benjamin Markines et al. "Evaluating Similarity Measures for Emergent Semantics of Social Tagging." In: *WWW*. 2009.
[18] Peter Mika. "Ontologies Are Us: A Unified Model of Social Networks and Semantics." In: *Web Semant.* 5.1 (Mar. 2007).
[19] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." In: *HLT-NAACL*. 2013.
[20] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality." In: *NIPS*. 2013.
[21] Thomas Niebler et al. "Extracting Semantics from Unconstrained Navigation on Wikipedia." In: *KI – Künstliche Intelligenz* 30.2 (2016).
[22] Thomas Niebler et al. "How Tagging Pragmatics Influence Tag Sense Discovery in Social Annotation Systems." In: *ECIR*. 2013.
[23] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global Vectors for Word Representation." In: *EMNLP*. Vol. 14. 2014.
[24] Bryan Perozzi, Rami Al-Rfou', and Steven Skiena. "DeepWalk: online learning of social representations." In: *KDD*. Ed. by Sofus A. Macskassy et al. 2014.
[25] Kira Radinsky et al. "A Word at a Time: Computing Word Relatedness Using Temporal Semantic Analysis." In: *WWW*. 2011.
[26] H. Schütze and J.O. Pedersen. "A cooccurrence-based thesaurus and two applications to information retrieval." In: *IPM* 33.3 (1997).
[27] Philipp Singer et al. "Computing Semantic Relatedness from Human Navigational Paths: A Case Study on Wikipedia." In: *IJSWIS* 9.4 (2013).
[28] Jian Tang et al. "LINE: Large-scale Information Network Embedding." In: *WWW*. 2015.
[29] Thomas Vander Wal. *Folksonomy Definition and Wikipedia*. Nov. 2005.
[30] Arkaitz Zubiaga et al. "Harnessing Folksonomies to Produce a Social Classification of Resources." In: *IEEE Trans. on Knowl. and Data Eng.* 25.8 (Aug. 2013).