Learning Semantic Relatedness from Human Feedback Using Metric Learning

Thomas Niebler[1*], Martin Becker[1], Christian Pölitz[1], and Andreas Hotho[1,2]

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

Abstract Assessing the degree of semantic relatedness between words is an important task with a variety of semantic applications, such as ontology learning for the Semantic Web, semantic search or query expansion. To accomplish this in an automated fashion, many relatedness measures have been proposed. However, most of these metrics only encode information contained in the underlying corpus and thus do not directly model human intuition. To solve this, we propose to utilize a metric learning approach to improve existing semantic relatedness measures by learning from additional information, such as explicit human feedback. For this, we argue to use word embeddings instead of traditional high-dimensional vector representations in order to leverage their semantic density and to reduce computational cost. We rigorously test our approach on several domains including tagging data as well as publicly available embeddings based on Wikipedia texts and navigation. Human feedback about semantic relatedness for learning and evaluation is extracted from publicly available datasets such as MEN or WS-353. We find that our method can significantly improve semantic relatedness measures by learning from additional information, such as explicit human feedback. For tagging data, we are the first to generate and study embeddings. Our results are of special interest for ontology and recommendation engineers, but also for any other researchers and practitioners of Semantic Web techniques.

1 Introduction

Automatically assessing semantic relatedness as perceived by humans is a task with many applications related to semantic technologies, such as ontology learning for the Semantic Web, semantic search or query expansion. Recent work has shown that semantic relatedness between words can successfully be extracted from a wide range of sources, such as tagging data [6, 17], Wikipedia article texts [13, 23] or Wikipedia navigation [20, 25]. In particular, such approaches usually encode semantic information of words in continuous *word vectors* [27]. The semantic relatedness between two words can then be measured by the cosine similarity of their corresponding word vectors. While the above-cited methods come close to human intuition of semantic relatedness, they are only able to encode information contained in the underlying corpus. Thus, they do not explicitly represent the actual notion of semantic relatedness as expected and employed by humans. The natural way to solve this problem is to incorporate additional information, such as explicit human feedback, in order to account for the deviations of the respective semantic relatedness measure from

human intuition. Furthermore, such feedback could be helpful to adapt semantic relatedness measures to specific domains and tasks or to personalize them in order to improve recommendation approaches or retrieval methods for search engines.

Problem Setting and Approach. Consequently, this work addresses the issue of incorporating additional information such as explicit human feedback about semantic relatedness into relatedness measures operating on vector representations of words. To this end, we apply a metric learning approach where we encode the additional information in the form of constraints. This manipulates the original relatedness measure and ultimately yields a better fit with human intuition.

There are many domains from which semantic relatedness has been extracted. Thus, we aim to propose a universally applicable approach. To illustrate the flexibility of this method, we apply it to Wikipedia article *texts*, *navigational traces* on the Wikipedia page network, and *tagging data*. To represent words in these application domains we use word embeddings, i.e., low-dimensional, dense vector representations, which reduce the computational complexity of our metric learning approach and have been shown to outperform high-dimensional representations in measuring semantic relatedness [1]. While word embedding approaches have been applied to several Wikipedia domains (texts or navigation) [8, 15, 21], we are, to the best of our knowledge, the first to derive tag embeddings and study the relationship between their dimensionality and their semantic expressiveness.

Independent of the domain, we show that our approach can significantly improve the quality of the given semantic relatedness measures. As mentioned earlier we confirm this on different domains by learning from and evaluating on a variety of well known semantic relatedness datasets generated from human intuition. In this context, we study the influence of the amount of information used for learning and investigate if the improved semantic relatedness measures generalize between different human intuition datasets.

Contribution. Our contribution is twofold: First, we introduce the metric learning setting (with relative constraints) to the domain of semantic relatedness and — by exploiting human feedback — show that it is possible to use metric learning to improve semantic relatedness measures to better fit human intuition. This also opens a new connection for the field of semantic relatedness research to a popular field in machine learning and can lead to another fruitful combination of both. We explicitly show how to adopt the metric learning scenario for our relatedness learning setting. Secondly, we are the first to generate and study embeddings from tagging data, which allows for effectively and efficiently performing metric learning.

Overall, our work describes a way to improve semantic relatedness measures based on additional semantic information, e.g., explicit human feedback. This enables us to increase the fit of these measures to human intuition significantly and even introduce user-specific information into the corresponding semantics. Our results are of special interest for ontology and search engineers, but also for any other practitioners of Semantic Web techniques.

Structure. The rest of this paper is structured as follows: in Section 2, we introduce our approach of learning semantic relatedness using metric learning. In Section 4, we describe the conducted experiments and present the results, which we discuss in Section 5. Section 6 gives an overview of other work related to this paper. Finally, Section 7 concludes this work and gives directions for future research.

2 Metric Learning to Learn Semantic Relatedness

In this section, we first formulate the goal of learning semantic relatedness in terms of metric learning. We then argue that the notions of distance and semantic relatedness are equivalent when restricting the setting to a transformed unit sphere. Finally, we introduce our method to formulate human feedback as constraints for the metric learning algorithm we employ. Figure 1 shows a sketch of the steps that we apply in our approach to learn semantic relatedness.

Learning Semantic Relatedness in Terms of Metric Learning. To learn a metric, standard metric learning algorithms parameterize the Mahalanobis distance, $d_M(x,y) = \sqrt{(x-y)^T M(x-y)}$, by finding a (symmetric, positive definite) matrix M. To this end, most algorithms expect a set of constraints $\mathcal{C}$. In this work, we apply the LSML algorithm [16], which learns from relative distance constraints of the form $\mathcal{C} := \{(x, x', y, y') : d(x, x') < d(y, y')\}$.

However, instead of learning a distance $d_M(x,y) : \mathbb{R}^n \times \mathbb{R}^n \to [0;\infty)$ we want to learn a *semantic relatedness* measure $rel_M(x,y) : \mathbb{R}^n \times \mathbb{R}^n \to [-1;1]$. Accordingly, instead of formulating constraints based on a distance d, they are based on a relatedness measure rel, i.e., in our case the cosine (cos). Overall, this amounts to learning a symmetric, positive definite matrix M such that the parametrized cosine measure $\cos_M(x,y) := x^T My \cdot (\|x\|_M \|y\|_M)^{-1}$, where $\|x\|_M := \sqrt{x^T Mx}$, suffices all constraints $\mathcal{C}$ which are derived from human intuition on semantic relatedness (instead of distance). In the following, we show that — when restricting the setting to a transformed unit sphere — learning a distance measure and a semantic relatedness measure is equivalent when specifying the constraints correctly.

Equivalence of Distance and Relatedness for Metric Learning. In the following, we first show the equivalence of distance and relatedness (as measured by the Mahalanobis metric and the parameterized cosine measure, respectively) on the transformed unit sphere $\mathbb{S}_M^{n-1} := \{x \in \mathbb{R}^n | \|x\|_M = 1\}$. This allows us to formulate constraints in a way to learn semantic relatedness.

For all vectors on the transformed unit sphere $x, y \in \mathbb{S}_M^{n-1}$ parametrized by M, the Mahalanobis distance metric and the parameterized cosine measure $cos_M(x,y)$

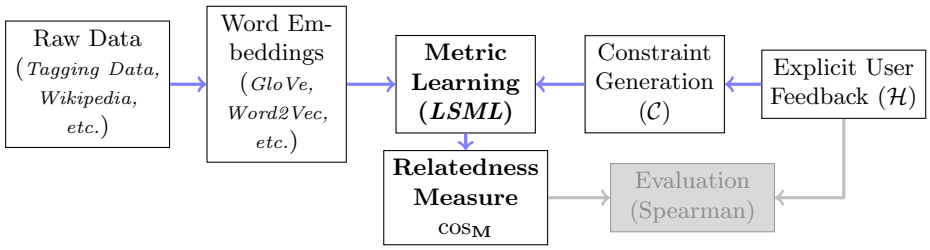


Figure 1: The pipeline of our approach. We preprocess raw data in order to create word embeddings, which serve as vector input for the metric learning algorithm. Simultaneously, we transform a portion of the user feedback information $\mathcal{H}$ to relatedness constraints $\mathcal{C}$ for the metric learning algorithm. The output of the algorithm is a relatedness measure $\cos_M$, characterized by a symmetric, positive-definite matrix M. Later, this matrix together with a portion of the explicit user feedback is used for the evaluation, which is explained in Section 4.1.

can be expressed as each other. That is, since $x^T Mx = 1$ and because of M being symmetric ($x^T My = y^T Mx$), it holds that: $d_M^2(x, y) = 2 \cdot (1 - \cos_M(x, y))$. This is in line with the intuition that if two words are more closely related, the distance of their vector representations is lower and vice versa. Thus, we can directly apply the metric learning approach to learning semantic relatedness measures on word vectors, if the constraints are correctly specified. To this end, we use the fact that the previous equation trivially implies:

$$d_M(x, x') < d_M(y, y') \Leftrightarrow \cos_M(x, x') > \cos_M(y, y')$$

Thus, we can formulate the constraints based on a relatedness notion rel instead of a distance d by specifying $\mathcal{C} := \{(x, x', y, y') : rel(x, x') > rel(y, y')\}$, i.e., the comparison operator is inverted.

Obtaining Relatedness Constraints from Human Feedback. To obtain suitable constraints to train the metric learning algorithm for learning semantic relatedness, we propose to exploit human intuition datasets. Such datasets contain word pairs together with human-assigned relatedness scores (see also Section 3.3) which can be interpreted as explicit human feedback. Formally, such datasets can be expressed as a set $\mathcal{H} := \{(w_i, w_i', r_i)\}$, where w_i and w_i' are words and r_i is the human-assigned score which describes an intuitive notion of the degree of relatedness between the two corresponding words. As such relatedness scores are commonly collected in a crowdsourcing task [5, 12], they thus represent explicit human feedback. To obtain constraints in the form (x, x', y, y') to use in the metric learning algorithm, one can simply combine all pairs of examples $(w_i, w_i', r_i), (w_j, w_j', r_j) \in \mathcal{H}$ with $r_i < r_j$ and thus receive a set of relatedness constraints $\mathcal{C} = \{(w_i, w_i', w_j, w_j')\}$.

Since we want to emphasize the importance of some constraints in the optimization step of the algorithm, we place higher weights on those constraints to make sure that they are fulfilled. The LSML algorithm allows to assign weights to all constraints. In order to put a high emphasis on a constraint with one very unrelated pair of words, e.g., (w_i, w_i', r_i), and another very related pair of words, e.g., (w_j, w_j', r_j) with $r_i \ll r_j$, we can define the weight of this constraint according to the difference of the respective human relatedness scores of the corresponding word pairs. The extended constraints can then be written as $\mathcal{C}_{weighted} := \{(w_i, w_i', w_j, w_j', r_j - r_i)\}$. In the remainder of this work, we always employ this kind of weighted constraints.

3 Datasets

In this work, use two different kinds of datasets to evaluate our metric learning approach to integrate user feedback to relatedness measures. That is, *domain datasets* which provide a set of word vectors representing the words to calculate semantic relatedness for, and *human intuition datasets* (HIDs) which we employ to learn semantic relatedness and to test our results. In the following we first describe two domain datasets containing tagging data from which we later derive tag embeddings. Then we review two domain datasets based on Wikipedia which come with pre-trained word vectors. Finally, we introduce all human intuition datasets containing human-assigned scores of similarities to word pairs.

3.1 Tagging Datasets to Derive Word Embeddings

In our work, we study datasets of two public social tagging systems. We use data from BibSonomy, which has a more academic audience. The second dataset is a subset of the Delicious social tagging system, where the audience is focused on design and technical topics.

Each dataset is restricted to the top 10k tags to reduce noise. Additionally, we only considered those tags from users who have tagged at least 5 resources and only those 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 [3]. We use a freely available dump of BibSonomy, covering all tagging data from 2006 till the end of 2015.[3] After filtering, it contains 10,000 distinct tags, which were assigned by 3,270 users to 49,654 resources in 630,955 assignments.

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 [33].[4] 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.

3.2 Pre-trained Embedding Datasets Based on Wikipedia

In order to demonstrate the applicability of our approach on any kind of word embeddings, we also use two publicly available datasets of pre-trained vectors. Both are related to Wikipedia, which has been shown time after time to yield high quality semantic content [8, 13, 20, 23, 25].

WikiGloVe. The authors of the GloVe embedding algorithm [21] trained several datasets of vector embeddings on various text data and made them publicly available.[5] Because it has been demonstrated several times that the textual content of Wikipedia articles can be exploited to calculate semantic relatedness [13, 23], we use the vectors based on Wikipedia as a reference for word embeddings generated from natural language. This dataset consists of 400,000 vectors with dimension 100.

WikiNav. Wulczyn published a set of word embeddings generated from navigation data on the Wikipedia webpage [30] using Word2Vec [18]. Word2Vec was originally intended to be applied on natural language text, though it can also be applied on navigational paths [8]. While technically the generated embeddings represent pages in Wikipedia, most pages also describe a specific concept and can thus be used interchangeably. It has been shown that exploiting human navigational paths as a source of semantic relatedness yields meaningful results [20, 25, 29]. The dataset at hand consists of 1,828,514 vectors with 100 dimensions. The vector embeddings

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

[4] http://www.zubiaga.org/datasets/socialbm0311/

[5] https://nlp.stanford.edu/projects/glove/

Table 1: Overview of all Human Intution Datasets (HIDs). For each HID, we give the number of word pairs, the number of unique words and the number of judgments per word pair. Also, for each embedding dataset, we give the number of matchable pairs, where both words are present in the dataset's vocabulary.

dataset	pairs	words	matches			
			BibSonomy	Delicious	WikiNav	WikiGloVe
Bib100	100	122	100	94	42	98
MEN	3000	751	465	1376	1227	3000
WS-353	353	437	158	202	173	353

have been created from all navigation data in the month of January 2017 and are publicly available.[6]

3.3 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, so called human intuition datasets (HIDs). They will provide training as well as test data. In the following, we will describe all used HIDs briefly. Table 1 gives an overview of the dataset sizes and the overlap as well as the Spearman correlation for the matchable pairs for all embedding datasets.

WS-353. The WordSimilarity-353[7] dataset consists of 353 pairs of English words and names [12]. 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 [5] contains 3,000 word pairs together with human-assigned similarity judgments, obtained by crowdsourcing using Amazon Mechanical Turk[8]. 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 more research and computer science oriented tagging data that we investigate.[9] It consists of 122 words from the top 3,000 words of the BibSonomy dataset and combined them into 100 word pairs, which subsequently were judged 26 times each for semantic relatedness using crowdsourced scores between 0 (no similarity) and 10 (full similarity).

[6] https://figshare.com/articles/Wikipedia_Vectors/3146878

[7] http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/wordsim353.html

[8] http://clic.cimec.unitn.it/~elia.bruni/MEN

[9] http://www.dmir.org/datasets/bib100

4 Experimental Setup and Results

In this section, we perform several sets of experiments in order to demonstrate the usefulness of metric learning for learning semantic relatedness. First, we describe how we evaluate the quality of a learned semantic relatedness measure. Then we study the procedure of generating word embeddings from tagging data and perform a qualitative evaluation. Finally, we train several metrics on a range of domains considering different amounts of user feedback, investigate whether it is possible to transport trained semantic knowledge across different collections of user feedback and finally assess the robustness of the learned semantic relatedness measures. We publish our code to enable reproducibility of our experiments.[10]

4.1 Evaluating the Quality of Semantic Relatedness Measures

Most of the time, the quality of semantic relatedness measures is assessed by how well it fits human intuition [13, 18, 25]. Human intuition is collected in Human Intuition Datasets (HID) as introduced in Section 3.3. The most widely-used method to evaluate semantic relatedness on such datasets is the Spearman rank correlation coefficient which compares the ranking of word pairs given by a HID with the ranking implied by the semantic relatedness measure. While there exist other evaluation approaches like analogy matching [15, 18] or concept categorization [1], they do not fit our setting, because we exclusively want to improve measuring relatedness.

4.2 Word Embeddings from Tagging Data

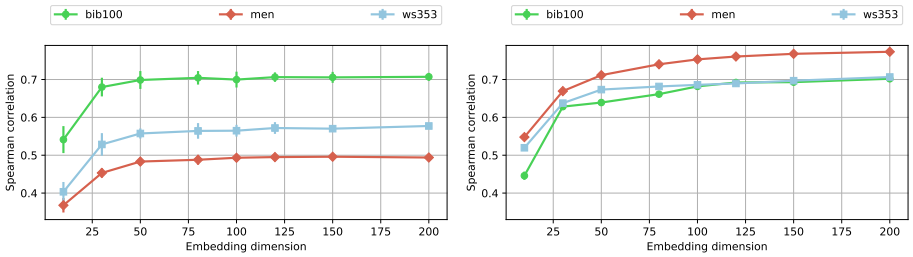
We evaluate our approach on several domains. This includes semantic relatedness extracted from tagging data. However, vector representations of words extracted from tagging data are traditionally high-dimensional [6, 17], making metric learning in this domain infeasible due to a more than quadratic runtime with regard to the number of vector dimensions [16]. Thus, similar to our Wikipedia examples we employ the notion of (low-dimensional) word embeddings which have been shown to outperform their high-dimensional counterparts in terms of correlation with human intuition of semantic relatedness [1]. This can also be confirmed when using tagging data as input (see Table 2, cf. ρ_{high} vs. ρ_{emb}). In this section, we justify our choice of using GloVe to embed words based on tagging data and study the influence of dimensionality on the respective semantic content.

Choosing an Embedding Algorithm. In this work, we apply the GloVe algorithm [21], which learns word embeddings from a word co-occurrence matrix. Other candidates are the well-known Word2Vec approach by [18] and the LINE algorithm [26]. However, Word2Vec relies on the meaningfulness of the sequential order of words which is not available from tagging data. LINE — which learns node embeddings preserving the first and second order neighborhood of the nodes in the graph — is more applicable. However, we found that it has a tendency to perform even worse than standard high dimensional representation for calculating semantic relatedness. Thus, overall we only report results on GloVe, since it was directly applicable to tagging data and showed the best results in our experiments.

[10] `http://dmir.org/semmele`

Table 2: Spearman correlation scores for both the high-dimensional representation (ρ_{high}) and word embeddings (ρ_{emb}) of both tagging datasets. It can be seen that the word embeddings encode semantic relations which are more in line with human judgment than the high-dimensional representations.

datasets	BibSonomy		Delicious	
	ρ_{high}	ρ_{emb}	ρ_{high}	ρ_{emb}
Bib100	0.621	0.726	0.640	0.675
MEN	0.436	0.483	0.581	0.752
WS-353	0.395	0.575	0.454	0.690

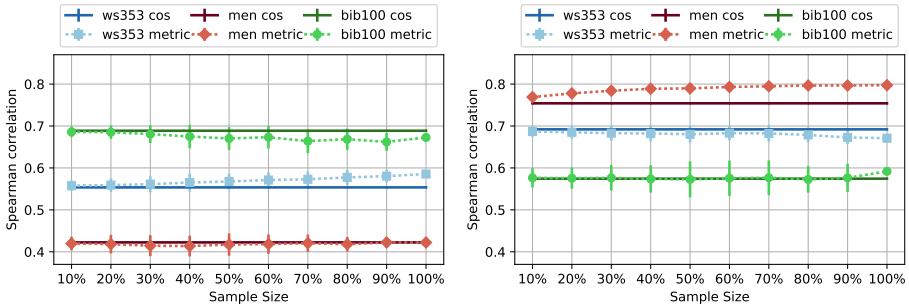


(a) BibSonomy (b) Delicious

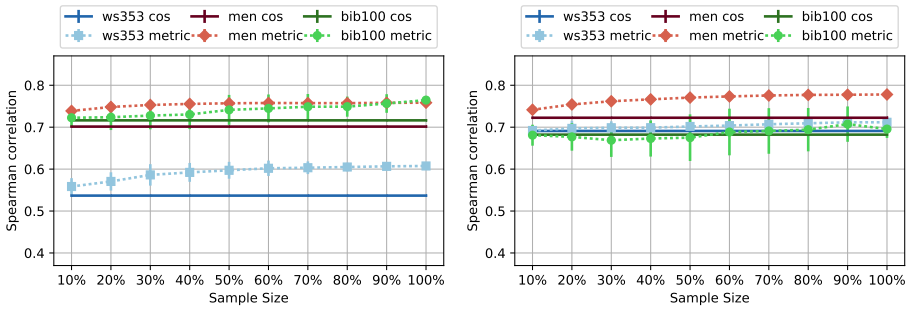
Figure 2: Impact of the vector dimension and random initialization of the embedding algorithm on the evaluation result on different HIDs across several vector dimension settings. The error bars show the standard deviation, the dots depict the mean of the evaluation results across 10 runs with the same parameter settings.

Embedding Dimension. One decision to make when generating word embeddings is choosing their dimension: We want the number of dimensions to be small to reduce the complexity of the metric learning approach, but it needs to be large enough to encode the necessary semantic information. In order to find a good embedding representation of the tags, we experimented with the dimension of the generated vector embeddings on the Delicious and BibSonomy tagging data measuring semantic relatedness using the standard cosine measure. Due to the internal random initialization of GloVe, we ran the vector embedding generation process 10 times for each number of dimensions in order to study the corresponding standard deviations.

The results for both experiments are shown in Figure 2. For BibSonomy the influence of the random initialization of GloVe on the semantic content of the vectors decreases with increasing dimensionality, as indicated by the error bars. For the larger and denser Delicious dataset, there is less room for the random initialization to influence the results. This explains the hardly visible standard deviations. With regard to the semantic content of the embeddings, the increase in semantic quality settles around a dimension of 100 for BibSonomy. For Delicious, adding more dimensions keeps increasing the semantic content. However, the 100 mark also signifies a drastic inhibition of the growth-rate. Thus, considering the quadratic training complexity of the LSML algorithm in terms of vector space dimension, we decided



(a) BibSonomy (b) Delicious



(c) WikiGloVe (d) WikiNav

Figure 3: Results on different levels of user feedback training data. The dashed lines show the mean Spearman correlations on the test data, using the trained metrics, together with the standard deviation of the results. The continuous lines report the Spearman values when applying the standard cosine on the test data.

to perform all of the following experiments on tagging data with 100-dimensional embeddings.

4.3 Integrating Different Levels of User Intentions

In this section we investigate how the amount of human feedback used for training influences the quality of the learned semantic relatedness measure. To this end, we evaluate various training set sizes extracted from the HIDs on the different embedding sets (pre-trained or extracted by GloVe, cf., Section 3).

For each HID, we first randomly sampled 20% of all matchable pairs as test sets. We gathered 5 such test datasets. For each test set, we sample training sets of different sizes (10% - 100%) on which we train a metric each. This metric is evaluated on the 20% of previously sampled test data. We repeat sampling training sets and learning 25 times. Then, for each training set size, we take the mean and the standard deviation over all experiments. As a baseline, we also report the Spearman correlation using the pure cosine measure on the test datasets. Figure 3 shows that we can indeed inject user feedback information about semantic relatedness into our

relatedness measure. The results indicate that we can best learn semantic relatedness with the human intuition encoded in the MEN dataset. This is consistent across all datasets except the BibSonomy embeddings. While it is also sometimes possible to use the knowledge from the WS-353 dataset to improve our semantic relatedness measure, it does not improve results as much as knowledge from other human intuition datasets. On the Delicious embeddings, it even decreases performance to learn from WS-353 knowledge. Surprisingly, while the Bib100 dataset yields the best results on the BibSonomy embeddings using the plain cosine measure, we cannot exploit the contained knowledge enough to learn semantic relatedness from it. Also, across all four embedding sets, the Bib100 dataset shows the biggest variance of results, while the standard deviation of the MEN dataset results are tightly bounded. We can generally observe that using vectors from WikiGloVe for training seems to be beneficial for our approach, as we can always improve the fit of our measure to human intuition significantly, regardless of the choice of training data.

4.4 Transporting User Intentions

The previous experiments showed that the integration of a dedicated HID into a relatedness measure results in higher agreement of the measure with human intuition. Now, in order to transfer different user intentions across different settings, we trained metrics on one complete HID and evaluated them on a different HID. For example, training was done using all WS-353 relations but the metric was evaluated on the MEN HID. By this we evaluate if the learned knowledge generalizes from one notion of semantic relatedness (represented by a specific HID) to another.

Results are given in Table 3. For each line, its header defines the dataset on which the metric was trained, while the column header is the dataset on which the trained metric was then evaluated. In each cell, the first value denotes the Spearman correlation of the cosine measure with the human relatedness scores in the evaluation dataset. The second value is the Spearman correlation of the relatedness scores calculated from the trained metric with the human relatedness scores in the evaluation dataset. Depending on whether the trained metric increased or decreased correlation with human intuition, we depict upwards or downwards arrows.

From Table 3, we can see some interesting results: Generally, training a metric on BibSonomy embeddings almost always yields bad transfer results. The only improvement using BibSonomy embeddings is on the WS-353 dataset, when evaluating the metric trained on MEN. The Delicious embeddings are only useful for improving relatedness scores when the trained metric is evaluated on the MEN dataset. The most interesting part of these results is that, using WikiGloVe embeddings, we can always increase correlation with human intuition by a notable margin. However, on the WikiNav embeddings, the results differ only by smaller margins. The only exception here is a notable improvement on the Spearman correlation value of WS-353, when using a metric trained on MEN data. Overall, it is generally possible to transfer knowledge from one HID to another. However, this is highly dependent on the underlying word representations.

Table 3: Results for user intention transport experiments. We trained a metric on all word pairs from the dataset given at the start of each line and evaluated them on the dataset given in the column header. The first value is the Spearman correlation for the cosine measure on the evaluation dataset, the second value is the Spearman value for the trained metric. The arrow denotes if we could transfer relevant information from one dataset to another or not.

(a) BibSonomy

	MEN	WS-353	Bib100
MEN	-	0.576↗0.591	0.726↘0.673
WS-353	0.484↘0.475	-	0.726↘0.687
Bib100	0.484↘0.462	0.576↘0.557	-

(b) Delicious

	MEN	WS-353	Bib100
MEN	-	0.690↘0.682	0.676↘0.644
WS-353	0.752↗0.766	-	0.676↘0.652
Bib100	0.752↗0.772	0.690↘0.679	-

(c) WikiGloVe

	MEN	WS-353	Bib100
MEN	-	0.533↗0.604	0.658↗0.726
WS-353	0.693↗0.729	-	0.658↗0.700
Bib100	0.693↗0.727	0.533↗0.601	-

(d) WikiNav

	MEN	WS-353	Bib100
MEN	-	0.729↗0.751	0.738↘0.737
WS-353	0.709↘0.703	-	0.738↘0.715
Bib100	0.709↗0.715	0.729↘0.718	-

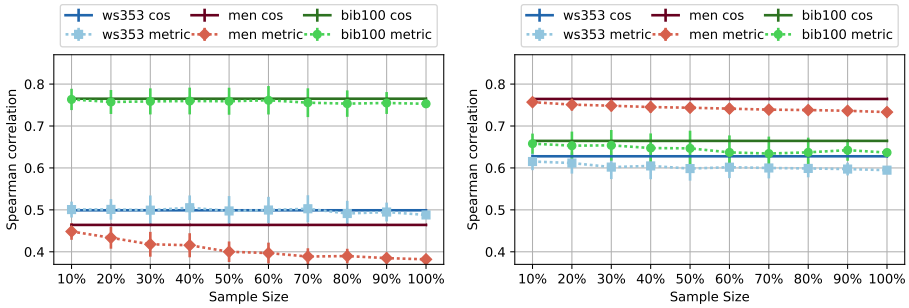
4.5 Robustness of the Learned Semantic Relatedness Measure

Here we inject wrong semantic relatedness information into our learning process. The goal is to show that i) wrong ratings do not collapse the relatedness measures, which ultimately makes our approach robust for different users with different intuitions of relatedness, and ii) that the promising results of the previous experiments are indeed caused by the successful injection of user feedback.

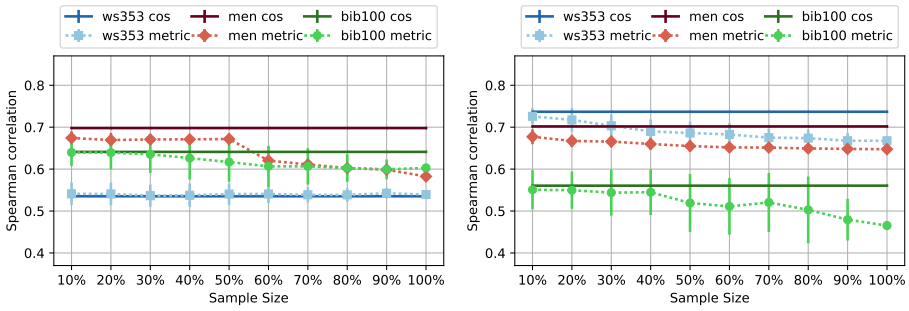
The setup is the same as with the random sampling experiment (Section 4.3), except that we randomly reassign the relatedness scores of the training pairs. This way, we evaluate on valid human intuition, but learn from false information. In Figure 4, we can see that shuffled relatedness scores exhibit a negative influence on the learned metric relatedness measure, as expected. On BibSonomy embeddings, only the measures trained on the MEN dataset yielded increasingly bad results, while the measures trained on the Bib100 or WS-353 datasets did not change very much, though they did not improve correlation either. All measures trained on Delicious embeddings dropped in performance. WS-353-trained measures stayed bad and showed even worse results, Bib100-trained measures also decreased in performance. Most notably, all embedding datasets except BibSonomy seem to be very receptive for changes induced by MEN relations: While performance increased in the first experiment, here, it decreased by a notable margin. Nevertheless, the decrease of all tested measures is mitigated by the inherent semantic content of the embeddings. Overall, this shows the robustness as well as the consistency of our approach, as was the goal of this experiment.

5 Discussion of the Results

Integrating Different Amounts of User Feedback. It can be seen that in the random sampling experiments, the information in the MEN dataset is generally best



(a) BibSonomy (b) Delicious



(c) WikiGloVe (d) WikiNav

Figure 4: Results for the robustness experiment. For each split, the scores of the word pairs in the training dataset were shuffled. The test dataset stayed the same.

suited to learn semantic relatedness. While this might be due to its big size, it might also be due to this dataset was created[11]: Using crowdsourcing, each human worker was shown two pairs of words and had to determine which of both pairs is more related. The higher a word pair in MEN is rated, the more often it was considered more related than the other one that was given. Our approach exploits very similar constraints for learning. Keeping this explanation in mind, we are able to give a recommendation on how to gather human feedback in order to learn semantic relatedness with our method. The bad performance of MEN on the BibSonomy embeddings, both with the baseline and the metric, could be attributed to the very low overlap of the BibSonomy vocabulary and the MEN pairs (see Table 1) as well as the small size of the BibSonomy tagging data. On all other embedding datasets, where MEN is very useful to learn the metric, the pair overlap is notably higher. Throughout all settings in this experiment, we observed that using more data results in better relatedness scores, if there was a positive effect.

Transporting User Intentions. The most notable result here is that knowledge transfer from one HID to another works best and with large improvements on the WikiGloVe embeddings. These embeddings are generated from the by far biggest

[11] `https://staff.fnwi.uva.nl/e.bruni/MEN`

word collections, i.e., the Wikipedia of 2014 as well as the Gigaword5 corpus.[12] It thus seems plausible that the embedding vectors encode a large portion of the semantic information of the underlying corpus and thus benefit most from the injected knowledge in our approach. It also shows that we are able to adapt the metric and transfer the knowledge if the vectors representations contain all necessary information. This is also in line with the notion that BibSonomy is the smallest corpus in our collection, which seems too sparse to properly learn semantic relatedness. Table 3a shows almost only deteriorating correlation scores when applying a learned relatedness measure, except for the MEN-trained measure evaluated on WS-353. Finally, results on WikiNav do not change very much, except when training the measure on MEN and evaluating it on WS-353, which is a similar phenomenon as on the BibSonomy embeddings.

Robustness. On all four embedding datasets, evaluation performance decreases notably with the MEN dataset, with the worst performance loss on BibSonomy, where MEN does not perform well anyway. We observe similar responses on WikiGloVe. These results confirm (again) that word embeddings successfully manage to encode semantic information, and also that we cannot just "unlearn" it. Furthermore, all embedding sets except BibSonomy react the most when used with a measure trained on MEN. We attribute this to the same reasons as why MEN is seemingly best suited to learn semantic relations from, i.e., it is constructed in a very similar way to the form of the constraints that the learning algorithm is parameterized with. Another consequence of this is that the promising results of the previous experiments are indeed caused by the successful injection of semantic side information into the relatedness measure.

Additional remarks. We are well aware that with our current set of vector embeddings, we do not improve upon the current state-of-the-art evaluation results on WS-353 and MEN. However, this was not our goal in this work, as we wanted to demonstrate the feasibility of our metric learning approach to inject prior knowledge from human feedback into a semantic relatedness measure.

6 Related Work

In the following, we will report the most relevant work in these fields of metric learning as well as semantic relatedness learning algorithms.

Metric Learning. Since we focus on the adaptation of metric learning on semantic relatedness constraints, we give a short overview of different types of metric learning algorithms. Metric learning algorithms can be roughly split in two classes according to the nature of the exploited constraints.

The first class of metric learning algorithms utilizes link-based constraints, i.e., we have explicit information if two items are either similar or dissimilar. As one of the first to propose an approach to learn a distance metric, Xing et al. [31] proposed to parameterize the Euclidean metric with a Mahalanobis matrix M in order to improve kNN clusterings by incorporating side knowledge. Weinberger et al. [28] presented the LMNN algorithm, which aims to improve kNN clustering by placing

[12] https://nlp.stanford.edu/projects/glove/

items with similar classes near to each other, while pushing away items with different classes by a large margin. The metric learning algorithm proposed in [9] makes use of quadruplets (x, x', y, y') and distance constraints u and l. These parameters can be translated to constraints $d(x, x') > u$ and $d(y, y') < l$. While the form of the constraints seems very similar to those of our approach, the constraints still only encode two classes of similar and dissimilar items. Finally, Qamar and Gaussier [22] propose an algorithm to learn a generalized cosine measure to improve kNN classification. This approach is similar to ours, as we also learn a generalized cosine measure, but their algorithm again exploits similarity and dissimilarity constraints with a large margin.

The other class of metric learning algorithms is based on relative constraints, e.g., for three items x, y, z, a constraint could be x is more similar to y than x to z. This setting is much more in line with the idea of actually measuring a continuous degree of relatedness instead of a preprocessing step for classification or clustering. In [24], Schultz and Joachims propose an early distance metric learning approach based on Ranking Support Vector Machines. Their algorithm takes sets of triplets (x_i, x_j, x_k) which encode the constraints $d(x_i, x_j) < d(x_i, x_k)$, i.e., x_i is more similar to x_j than to x_k. These constraints are extracted from clickthrough data, where explicit preference information of a list of items compared to a reference item is available. This is however not the case in our scenario, as our constraints are based on distance comparisons between four different items instead of only three. The algorithm proposed in [16] makes use of relative distance comparisons encoded in quadruplets (x, x', y, y'), which encode relative comparisons $d(x, x') < d(y, y')$ without a separation margin, in order to learn a metric. This is a more general approach than the one provided by [24], as it is easy to convert triplet constraints to quadruplet constraints, but not the other way round.

Learning Semantic Relatedness. While the task to correctly determine the semantic relatedness of words or texts has been around for a long time, there are still few approaches which actually learn semantic relatedness.

Lately, many unsupervised approaches to learn semantic relatedness in low dimensions have been proposed. These methods are also often called *word embedding algorithms*. Such methods train a model to predict a word from a given context [2, 7, 18, 26]. Other embedding methods focus on factorizing a term-document matrix [10, 21]. These methods all have in common that they do not inject any external knowledge. Anyhow, [1] showed that all those methods generally exhibit a notably higher correlation with human intuition than the standard high-dimensional vector representations proposed by [27].

Bridging the gap between unsupervised relatedness learning approach and human intuition by injecting side knowledge can be accomplished with post-processing methods or with directly injecting this knowledge in the embedding process. Both [14] and [19] propose approaches to inject synonymy and, in the case of the latter, also antonymy constraints into semantic vector representations. They aim to maximize the similarity of synonymous words, while minimizing the similarity of antonyms. Hereby, synonymy constraints acted as attractors in the semantic vector space, while the antonymy constraints acted as repellants. [11] presented a method to fit the embedding vectors to the neighborhood defined by relations in semantic lexicons. In a way, also this algorithm is based on similarity constraints, as the

distance between similar vectors is minimized. However, all of these works did not incorporate the actual degree of relatedness into their approaches, which is what we do in this work. There also exist methods which incorporate side knowledge directly into the embedding process, e.g., [4, 32]. However, our metric learning approach works on any already existing set of vector embeddings instead of actually training new word embeddings from raw data.

7 Conclusion

In this work, we presented an approach to learn semantic relatedness from human intuition based on word embeddings. Our approach is scalable and fast in terms of constraints and produces significantly improved results compared to the widely used cosine measure, while yielding competitive results on human evaluation datasets. We argued for the use of word embeddings instead of high-dimensional vector representations for tagging data due to an improvement in their semantic content and their clear reduction of computational complexity when learning a metric.

Concretely, we could show that we can exploit semantic relatedness information from HIDs to more realistically assess semantic relatedness, regardless of the underlying embedding dataset. Aditionally, we were able to encode and transfer knowledge from one HID to another, sometimes with a very large increase of correlation with human intuition. When training a metric on false information to assess the robustness of our approach, we argued that this actually supports our results, as the algorithm yields negative results, as we expected. Transferred to our previous positive results, we are indeed able to inject valid knowledge into our relatedness measure to produce a better fit to human intuition than only with word embeddings.

Future work includes the exploration of other graph embedding algorithms for tagging data, the exploration of crowdsourcing strategies to best gather data suitable for metric learning and further adaptation of metric learning algorithms to specific properties of social tagging systems.

Acknowledgements. This work has been partially funded by the DFG grant "Posts II" and the BMBF funded junior research group "CLiGS" (grant identifier FKZ 01UG1408).

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. "The Social Bookmark and Publication Management System BibSonomy." In: *VLDB* (2010).

[4] Jiang Bian, Bin Gao, and Tie-Yan Liu. "Knowledge-powered deep learning for word embedding." In: *ECML/PKDD*. 2014.

[5] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. "Multimodal Distributional Semantics." In: *JAIR* (2014).

[6] Ciro Cattuto et al. "Semantic Grounding of Tag Relatedness in Social Bookmarking Systems." In: *ISWC*. 2008.

[7] Ronan Collobert and Jason Weston. "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning." In: *ICML*. 2008.

[8] Alexander Dallmann et al. “Extracting Semantics from Random Walks on Wikipedia: Comparing learning and counting methods.” In: *WikiWorkshop@ICWSM*. 2016.
[9] Jason V. Davis et al. “Information-theoretic metric learning.” In: *ICML*. 2007.
[10] Scott Deerwester et al. “Indexing by latent semantic analysis.” In: *Journal of the American Society for Information Science* 41.6 (1990).
[11] Manaal Faruqui et al. “Retrofitting Word Vectors to Semantic Lexicons.” In: *CoRR* (2014).
[12] Lev Finkelstein et al. “Placing Search in Context: the Concept Revisited.” In: *WWW*. 2001.
[13] Evgeniy Gabrilovich and Shaul Markovitch. “Computing semantic relatedness using Wikipedia-based explicit semantic analysis.” In: *IJCAI*. 2007.
[14] Guy Halawi et al. “Large-scale Learning of Word Relatedness with Constraints.” In: *KDD*. 2012.
[15] Omer Levy, Yoav Goldberg, and Israel Ramat-Gan. “Linguistic Regularities in Sparse and Explicit Word Representations.” In: *CoNLL*. 2014.
[16] E. Y. Liu et al. “Metric Learning from Relative Comparisons by Minimizing Squared Residual.” In: *ICDM*. Dec. 2012.
[17] Benjamin Markines et al. “Evaluating Similarity Measures for Emergent Semantics of Social Tagging.” In: *WWW*. 2009.
[18] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality.” In: *NIPS*. 2013.
[19] Nikola Mrkšić et al. “Counter-fitting Word Vectors to Linguistic Constraints.” In: *HLT-NAACL*. 2016.
[20] Thomas Niebler et al. “Extracting Semantics from Unconstrained Navigation on Wikipedia.” In: *KI - Künstliche Intelligenz* (2015).
[21] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global Vectors for Word Representation.” In: *EMNLP*. Vol. 14. 2014.
[22] Ali Mustafa Qamar and Eric Gaussier. “Online and batch learning of generalized cosine similarities.” In: *ICDM*. 2009.
[23] Kira Radinsky et al. “A Word at a Time: Computing Word Relatedness Using Temporal Semantic Analysis.” In: *WWW*. 2011.
[24] Matthew Schultz and Thorsten Joachims. “Learning a distance metric from relative comparisons.” In: *NIPS*. 2004.
[25] Philipp Singer et al. “Computing Semantic Relatedness from Human Navigational Paths: A Case Study on Wikipedia.” In: *IJSWIS* (2013).
[26] Jian Tang et al. “LINE: Large-scale Information Network Embedding.” In: *WWW*. 2015.
[27] Peter D. Turney and Patrick Pantel. “From Frequency to Meaning: Vector Space Models of Semantics.” In: *J. Artif. Int. Res.* 37.1 (Jan. 2010).
[28] Kilian Q. Weinberger and Lawrence K. Saul. “Distance Metric Learning for Large Margin Nearest Neighbor Classification.” In: *JMLR* (2009).
[29] Robert West, Joelle Pineau, and Doina Precup. “Wikispeedia: an online game for inferring semantic distances between concepts.” In: *IJCAI*. 2009.
[30] Ellery Wulczyn. *Wikipedia Navigation Vectors*. May 2016.
[31] Eric P Xing et al. “Distance metric learning with application to clustering with side-information.” In: *NIPS* (2003).
[32] Mo Yu and Mark Dredze. “Improving Lexical Embeddings with Semantic Knowledge.” In: *ACL (2)*. 2014.
[33] 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).