Retrieving Multi-Entity Associations: An Evaluation of Combination Modes for Word Embeddings

Gloria Feher
Heidelberg University
Heidelberg, Germany
gloria.e.feher@gmail.com

Andreas Spitz
Heidelberg University
Heidelberg, Germany
spitz@informatik.uni-heidelberg.de

Michael Gertz
Heidelberg University
Heidelberg, Germany
gertz@informatik.uni-heidelberg.de

ABSTRACT

Word embeddings have gained significant attention as learnable representations of semantic relations between words, and have been shown to improve upon the results of traditional word representations. However, little effort has been devoted to using embeddings for the retrieval of entity associations beyond pairwise relations. In this paper, we use popular embedding methods to train vector representations of an entity-annotated news corpus, and evaluate their performance for the task of predicting entity participation in news events versus a traditional word cooccurrence network as a baseline. To support queries for events with multiple participating entities, we test a number of combination modes for the embedding vectors. While we find that even the best combination modes for word embeddings do not quite reach the performance of the full cooccurrence network, especially for rare entities, we observe that different embedding methods model different types of relations, thereby indicating the potential for ensemble methods.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**; **Retrieval effectiveness**.

KEYWORDS

word embeddings; embedding vector combination; implicit network; entity network

ACM Reference Format:
Gloria Feher, Andreas Spitz, and Michael Gertz. 2019. Retrieving Multi-Entity Associations: An Evaluation of Combination Modes for Word Embeddings. In *42nd Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331366

1 INTRODUCTION

Word embeddings are learned dense vector representations of words, which encode information on word context. They have established themselves as a popular way to encode unstructured text due to their numerous useful properties, such as the clustering of semantically or syntactically related words in the vector space, or the support of arithmetic operations on word vectors to "calculate" word analogies. These characteristics lend themselves to tasks in natural language processing and information retrieval, where embeddings can be used to alleviate vocabulary mismatch [7], for example. Similarly, sensitivity classification [10] and large-scale text classification [1] have been improved by using embeddings, although only in combination with other task-relevant features. It is thus important to investigate when, how, and why word embeddings perform so well, and when they do not. In particular, the issue has been raised that word embeddings provide meaningless similarities between otherwise unrelated words if the entire vector space is considered [9], and it has been demonstrated that local embeddings outperform global embeddings for query expansion [5]. These considerations imply that the potential neighbourhood in the word embedding space is too large and needs to be restricted in order to mimic only meaningful relations, which raises some questions such as: (1) Do word embeddings universally capture the relevant associations encoded in language better than other methods? (2) How is the neighbourhood of embeddings best used to solve a task pertaining to non-trivial word associations?

To investigate these questions, we consider the task of event completion, where one held-out entity is predicted from the remaining entities that participate in an event. An entity is said to participate in an event, if it is named in its description. Thus, predicting one entity from other participating entities is a suitable problem to evaluate relevant associations between words as captured by word embeddings, as well as different combination modes of word vectors to exploit the neighbourhood relationships. Since the task relies on the cooccurrence of entities in a common context, it lends itself to the use of embeddings. However, the fact that an entity may occur in different contexts provides a challenge for learned word vectors (for example, Brazil held the Summer Olympics in 2016, and in the same year impeached its president Dilma Rousseff for breaking fiscal laws). Benchmarks and training data for such an event completion task are provided by Spitz and Gertz [22], who used it to evaluate ranking in entity cooccurrence networks. Research by Schnabel et al. suggests that there is no universally best embedding method since the performance of embeddings varies by task [17]. We therefore evaluate word2vec-CBOW, word2vec-skip-gram, and GloVe to determine which method is best suited for identifying the participation of entities in events, and discuss the benefits and drawbacks of each tested method.

Contributions. We make two primary contributions. (1) We compare six modes of combining embedding vectors for multi-entity queries. (2) Based on a comparison to entity ranking in cooccurrence networks, we discuss the influence of an entity's frequency in the corpus on the performance of its resulting embedding.

SIGIR '19, July 21–25, 2019, Paris, France

2 RELATED WORK

Word embeddings are distributional vector representations of words that have been researched since the 1990s [6], leading to methods such as Latent Dirichlet Allocation [3] and neural language models [2]. These provide the groundwork for current word embedding models, which usually employ shallow neural networks. Recently, the context-based embeddings ELMo [16] and BERT [4] have been introduced, which are generated by data and training intensive deep architectures. These word representations vary with the input sentence(s), and are thus unsuitable for an isolated entity retrieval task, so we rely on more traditional embeddings, which yield only one fixed vector per word.

Word2vec is one of the most widely used models and comes in two variants [11, 12]. Continuous bag-of-words (CBOW) averages the word vectors of context words and uses them to classify the focal word. Inversely, continuous skip-gram predicts the context words from the focal word. Thus, word vectors are learned from local context windows. Typically, word2vec is modified to include negative sampling [12], which distinguishes between the real context words and a sample that is drawn from a noise distribution.

GloVe was proposed as global word representation vectors that improve upon the approach of word2vec [15]. To this end, they make use of the global corpus statistics by training a log-bilinear regression model on the word cooccurrence matrix. By employing a more general weighting function, they also gain more control over how strongly frequent words influence the model.

Implicit networks, in contrast, were introduced to model latent relationships between entities as well as the remaining terms [20]. The latent relationships are captured by constructing an entity and term coocurrence graph that is weighted by cross-sentence distances within the documents, thus capturing all word cooccurrences in the text. While the model has a variety of applications like event extraction or entity-centric topic extraction [21], it comes with event completion ground-truth data for news articles [22], so we use it as a baseline. With the prevalence of machine learning applications, it is also of interest to evaluate how well the implicit network performs in comparison to learned representations.

Combining word vectors as a technique is typically employed for creating document embeddings from word embeddings [1, 14] or for modeling multi-word entities in knowledge bases [18]. Several combination modes exist in the literature, mainly component-wise minimum and maximum, as well as averaging [1, 18, 19], which are often combined by concatenation into a single vector. Another approach is summing over the similarities between multiple query vectors and document vectors [14]. Mitchell and Lapata compare several composition functions for handcrafted cooccurrence-based word vectors [13]. Iacobacci et al. compare word embedding combinations for the task of word sense disambiguation [8]. However, to the best of our knowledge, no previous publication has compared different combination modes of learned word vectors obtained from several embedding methods on an entity-centric task.

3 EXPERIMENTAL SETUP

We briefly describe the task and training data, as well as the necessary steps for tuning the used embeddings.

Event completion task. To formalize the task, we assume that an event is defined by its k participating entities of type location, person, or organization. For each event, we generate k queries by holding-out one entity and using the remaining $k-1$ entities as query input. The task is then to predict the held-out entity, i.e., given the query entities and the type of the target entity, each model should predict the held-out entity. In practice, we treat this as a ranking task in which we rank nearest neighbours by cosine distance (for embeddings) or adjacent nodes by edge weights (for the implicit network). For each query, we generate a ranking of target entities to calculate *precision*@1 and *recall*. For example, the event of the third presidential debate of the 2016 U.S. election between Hillary Clinton (HC) and Donald Trump (DT) at the University of Nevada, Las Vegas (UNLV), would yield the three queries $\langle\{\textsc{hc},\textsc{dt}\} \rightarrow \{\textsc{unlv}\}\rangle$, $\langle\{\textsc{hc},\textsc{unlv}\} \rightarrow \{\textsc{dt}\}\rangle$, and $\langle\{\textsc{dt},\textsc{unlv}\} \rightarrow \{\textsc{hc}\}\rangle$.

Data. Since event completion is most relevant on news data, we train all models on a corpus of 127,485 English news articles from June 2016 to November 2016 [22], and follow the preprocessing steps described there to construct the implicit network. To train the embeddings, we replace named entity mentions in the text with Wikidata IDs. As ground truth, we use data from the same source [22], which was accumulated by crawling the Wikipedia Current Events portal for events that are described in articles within the corpus, and removing events that contain less than two entities. For our evaluation, we also exclude all queries in which the target entity is missing in at least one of our evaluated models (due to different window sizes and retention threshold values, not all models contain all entities). The final ground truth contains 263 queries.

Parameter tuning. We conduct preliminary hyperparameter optimization for all embedding models and select the best settings per method according to the achieved *precision*@1 scores. All embeddings are trained for 100 epochs, and we consider embeddings of dimension 50, 100, and 200. Where not otherwise specified, we choose hyperparameters according to the recommended default values. Due to the non-deterministic nature of word2vec and GloVe, we train each embedding ten times and average the results. Training is time intensive and takes up to 18h per model on a dual-core machine for skip-gram models.

Multi-entity neighbourhood. To use embeddings to predict the target entity, we propose a new mode for combining the word vectors of individual query entities and test five further modes from the literature. Let Q denote the set of query entities. Let $x \in X$ be one out of all possible target entities, with θ_x denoting its word vector, and $\Theta^Q = [\theta_{q1}, \theta_{q2}, \cdots]$ the matrix of horizontally stacked word vectors of query entities. Then $t_{\text{MODE}} \in X$ is the predicted target entity according to the respective ranking functions.

$$t_{\text{MINMAX}} = \operatorname*{argmin}_{x \in X} \; \operatorname*{argmax}_{q \in Q} \; cosdist(\theta_q, \theta_x) \tag{1}$$

$$t_{\text{SUM}} = \operatorname*{argmin}_{x \in X} \sum_{q \in Q} cosdist(\theta_q, \theta_x) \tag{2}$$

$$t_{\text{AVG}} = \operatorname*{argmin}_{x \in X} \; cosdist\Big(\frac{1}{|Q|} \sum_{q \in Q} \theta_q, \theta_x\Big) \tag{3}$$

$$t_{\text{CWMIN}} = \operatorname*{argmin}_{x \in X} \; cosdist\Big([\min(\Theta^Q_1), \cdots, \min(\Theta^Q_{|Q|})]^T, \theta_x\Big) \tag{4}$$

$$t_{\mathrm{CWMULT}} = \operatorname*{argmin}_{x \in X} \; cosdist\Big(\theta_{q_1} \odot \cdots \odot \theta_{q_{|Q|}}, \quad \theta_x\Big), q_i \in Q \quad (5)$$

Thus, MINMAX (Eq. 1) finds the word vector with the minimal maximal distance to the query vectors, SUM (Eq. 2) finds the word vector with the minimal sum of cosine distances to all query vectors [14], and AVG (Eq. 3) first averages the query vectors and then finds the nearest neighbour [1]. We further denote finding the word vector with the smallest cosine distance to the component-wise minimum of all query vectors with CWMIN (Eq. 4) [1], the component-wise maximum with CWMAX (analogous to Eq. 4) [1] and the component-wise multiplication with CWMULT (Eq. 5) [13].

Note that these combination modes are especially suitable for applications that do not depend on word order or syntactical information, since all of the evaluated modes are symmetrical. The event completion task is agnostic to word order as it operates on sets of entities. This is also true for implicit networks, which establish entity relations based on (symmetric) sentence-distance.

4 EVALUATION

In order to compare the word vector combination modes, we first focus our evaluation on their performance in the event completion task. Figure 1 shows the *recall@k* for all combination modes and embedding methods. We observe that the combination modes perform similarly across the different embeddings, where both SUM and AVG perform comparably well in terms of recall, albeit worse than the implicit network baseline. MINMAX, CWMIN, CWMAX and CWMULT perform considerably worse and do not reach a recall above 0.6 at rank 10, unlike the other combination modes. In terms of *precision*@1, SUM significantly outperforms the other modes across all embedding methods, as is further highlighted in Table 1. We thus focus on SUM as a combination mode in the following.

In Figure 2, we show the *precision*@1 for the SUM mode and the embedding models that perform best out of all tested hyper-parameter settings, to compare embedding spaces with differing dimensionalities. For CBOW, we obtain the best performance for a down-sampling threshold of 10^{-5}, 15 negative samples, a context window size of 21, and a minimum of 3 word occurrences in the training data. Similarly, skip-gram performs best for a down-sampling threshold of 10^{-5}, and a window size of 21. Like the word2vec models, GloVe yields the best results for a window size of 21, as well as setting $\alpha = 0.6$ and $x_{max} = 25$ in the weighting function of the least squares objective [15].

Overall, we observe two general trends. First, increasing the embedding dimension almost universally improves the performance.

Table 1: Average $prc@1$ for the event completion task for all modes and all embedding models of size 200. The implicit network baseline achieves an average $prc@1$ of 0.330.

	skip-gram	CBOW	GloVe
SUM	0.257	0.234	0.252
AVG	0.140	0.116	0.101
MINMAX	0.189	0.186	0.168
CWMAX	0.140	0.102	0.085
CWMIN	0.130	0.095	0.095
CWMULT	0.085	0.066	0.056

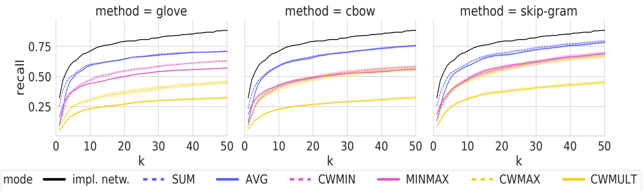

Figure 1: *Recall@k* for GloVe, CBOW, and skip-gram embeddings with the combination modes SUM, MINMAX, AVG, CWMIN, CWMAX and CWMULT. The implicit network is included as a baseline.

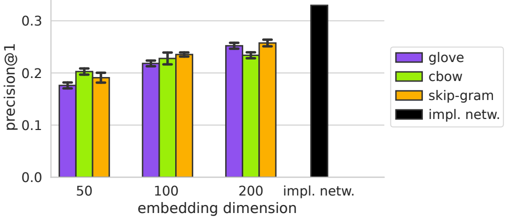

Figure 2: Average $prc@1$ for CBOW, skip-gram, and GloVe models with dimension 50, 100, 200 (using mode SUM).

Second, using a larger window size of 21 yields improvements across all methods (compared to the recommended values of 15 for GloVe, 10 for skip-gram, and 5 for CBOW). This indicates the importance of considering larger contexts to capture relevant relations between entities that are typically farther apart in a text, and is further corroborated by the implicit network's increased performance as it considers all relations within a document.

Another interesting aspect to consider is the frequency of target entities in the training data, since it should be more difficult to train models for rare entities. As the results in Figure 3 show, the implicit network unsurprisingly performs best for rare entities as it is constructed to retain the full, uncompressed cooccurrence information. However, while skip-gram and CBOW both benefit from higher entity frequencies, skip-gram initially performs best (even better than the implicit network) for extremely rare entities (frequency ≤ 10). GloVe produces the most curious results and performs worst for only some entities with very low frequencies, but also for those with mid-range frequencies, despite an overall performance that is close to skip-gram, as shown in Figure 2.

Due to the similar overall performance of GloVe and skip-gram in contrast to the apparent difference in performance depending on entity frequencies, we also investigate whether the models capture similar relations between entities in comparison to the implicit network. We randomly select 25 entities of each type and use the 100 most closely related entities in the implicit network as a pseudo ground-truth against which we rank the predictions that are generated by the embeddings. In Figure 4, we show the recall curves for the three embedding models. Note that the recall does not always reach a value of 1 since some rare entities are not contained in all embeddings due to the window size constraints that are more strict than in the implicit network. Here, we find that the word2vec models appear to rank entities in a similar manner to the implicit network, whereas GloVe ranks them differently and initially has lower recall. However, since GloVe performs similarly well on the

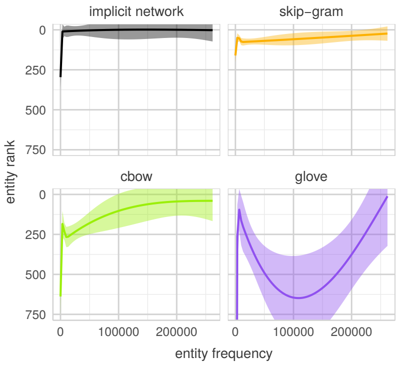

Figure 3: Obtained ranks of target entities for all methods vs. the frequency of target entities in the training corpus. Shaded areas denote 0.95 confidence intervals.

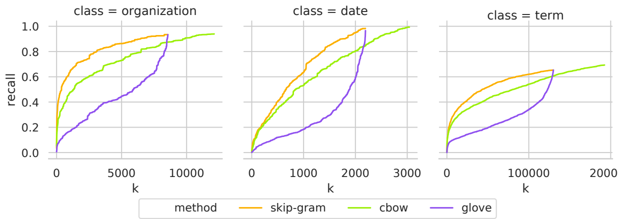

Figure 4: Average recall curves of predictions by all embedding methods (using SUM) for the top 100 neighbours of 25 randomly selected entities in the implicit network.

event completion task, this indicates that GloVe embeddings model a different sort of relation between the involved entities of events that nevertheless results in a similar overall performance.

5 CONCLUSION AND ONGOING WORK

For our investigation into the combination modes of word vectors, we conclude that summing over the cosine distances between the query vectors and prospective target vectors yields the highest precision and recall in the event completion task. While we focused on entities in our evaluation due to the availability of suitable training, query, and ground truth data, the experimental setup is generic. Therefore, we expect the results to be similar for multi-word associations in general once respective data sets become available.

Due to their compact representation and computational efficiency, learned word vectors are appealing for many applications. However, for the semantic-relatedness task of retrieving multi-entity associations, our experiments show that the representations obtained from word2vec and GloVe do not yet reach the performance of a full cooccurrence network representation, even though they are close. This, of course, raises the question why the learned embeddings do not attain the same performance level as the heuristic representation, and how this can be addressed in the future.

One potential explanation for the performance gap is the sparseness of entity mentions in comparison to general terms. Where the implicit network contains edges that represent cooccurrences over distances of multiple sentences, the embeddings are limited by a stricter window size, which conforms with their increased performance for increasing window size. However, increasing the window size does not scale arbitrarily due to computational restrictions on the runtime and the introduction of noise. While entities are bound to share some relation across sentence boundaries, this is less likely to be the case for arbitrary terms. Here, combination methods for training embeddings stand to improve the overall performance.

Our final observation concerns the fact that GloVe captures inherently different entity associations when compared to both word2vec and the implicit network. While implicit networks may generally capture more of the useful entity relations, this difference indicates that they apparently miss some of the associations that are emphasized in the global embedding process. A further analysis of these associations and how they occur would be of interest. Combining the network-based model with multiple learned features in an ensemble approach may benefit the overall performance, and requires a more in-depth investigation of the differences between the underlying relationships of entities in the different methods.

REFERENCES

[1] Georgios Balikas and Massih-Reza Amini. 2016. An Empirical Study on Large Scale Text Classification with Skip-gram Embeddings. *CoRR* abs/1606.06623 (2016).

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research* 3 (2003), 1137–1155.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018).

[5] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query Expansion with Locally-trained Word Embeddings. In *ACL'16*.

[6] Jeffrey L. Elman. 1991. Distributed Representations, Simple Recurrent Networks, and Grammatical Structure. *Machine Learning* 7 (1991), 195–225.

[7] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth JF Jones. 2015. Word Embedding Based Generalized Language Model for Information Retrieval. In *SIGIR'15*.

[8] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for Word Sense Disambiguation: An Evaluation Study. In *ACL'16*. 897–907.

[9] Jussi Karlgren, Anders Holst, and Magnus Sahlgren. 2008. Filaments of Meaning in Word Space. In *ECIR'08*.

[10] Graham McDonald, Craig Macdonald, and Iadh Ounis. 2017. Enhancing Sensitivity Classification with Semantic Features Using Word Embeddings. In *ECIR'17*.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS'13*.

[13] Jeff Mitchell and Mirella Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science* 34, 8 (2010), 1388–1429.

[14] Eric T. Nalisnick, Bhaskar Mitra, Nick Craswell, and Rich Caruana. 2016. Improving Document Ranking with Dual Word Embeddings. In *WWW'16 Companion*.

[15] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP'14*.

[16] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *NAACL-HLT'18*. 2227–2237.

[17] Tobias Schnabel, Igor Labutov, David M. Mimno, and Thorsten Joachims. 2015. Evaluation Methods for Unsupervised Word Embeddings. In *EMNLP'15*.

[18] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *NIPS'13*. 926–934.

[19] Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In *NIPS'11*.

[20] Andreas Spitz and Michael Gertz. 2016. Terms over LOAD: Leveraging Named Entities for Cross-Document Extraction and Summarization of Events. In *SIGIR'16*.

[21] Andreas Spitz and Michael Gertz. 2018. Entity-Centric Topic Extraction and Exploration: A Network-Based Approach. In *ECIR'18*.

[22] Andreas Spitz and Michael Gertz. 2018. Exploring Entity-centric Networks in Entangled News Streams. In *WWW'18 Companion*.