Evaluating Word Embeddings
in Multi-label Classification
Using Fine-grained Name Typing

Yadollah Yaghoobzadeh[1]   **Katharina Kann**[2]   **Hinrich Schütze**[3]

[1]Mirosoft Research, Montreal, Canada
[2]Center for Data Science, New York University, USA
[3]CIS, LMU Munich, Germany
`yayaghoo@microsoft.com`

Abstract

Embedding models typically associate each word with a single real-valued vector, representing its different properties. Evaluation methods, therefore, need to analyze the accuracy and completeness of these properties in embeddings. This requires fine-grained analysis of embedding subspaces. Multi-label classification is an appropriate way to do so. We propose a new evaluation method for word embeddings based on multi-label classification given a word embedding. The task we use is fine-grained name typing: given a large corpus, find all types that a name can refer to based on the name embedding. Given the scale of entities in knowledge bases, we can build datasets for this task that are complementary to the current embedding evaluation datasets in: they are very large, contain fine-grained classes, and allow the direct evaluation of embeddings without confounding factors like sentence context.

1   Introduction

Distributed representation of words, aka word embedding, is an important element of many natural language processing applications. The quality of word embeddings is assessed using different methods. Baroni et al. (2014) evaluate word embeddings on different intrinsic tests: similarity, analogy, synonym detection, categorization and selectional preference. Different concept categorization datasets are introduced. These datasets are small (<500) (Baroni et al., 2014; Rubinstein et al., 2015) and therefore measure the goodness of embeddings by the quality of their clustering. Usually cosine is used as the similarity metric between embeddings, ignoring subspace similarities.
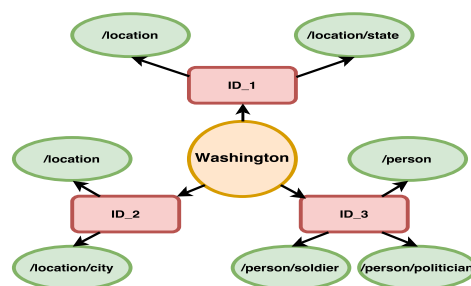

Figure 1: Types (ellipses; green) of the entities (rectangles; red), to which the name "Washington" can refer. Ideally, the embedding for "Washington" should represent all these types.

Extrinsic evaluations are also used, cf. Li and Jurafsky (2015). In these tasks, embeddings are used in context/sentence representations with composition involved.

In this paper, we propose a new evaluation method. In contrast to the prior work on *intrinsic evaluation*, our method is supervised, large-scale, fine-grained, automatically built, and evaluates embeddings in a classification setting where different subspaces of embeddings need to be analyzed. In contrast to the prior work on *extrinsic evaluation*, we evaluate embeddings in isolation, without confounding factors like sentence contexts or composition functions.

Our evaluation is based on an entity-oriented task in information extraction (IE). Different areas of IE try to predict relevant data about entities from text, either locally (i.e., at the context-level), or globally (i.e., at the corpus-level). For example, local (Zeng et al., 2014) and global (Riedel et al., 2013) in relation extraction, or local (Ling and Weld, 2012) and global (Yaghoobzadeh and Schütze, 2015) in entity typing. In most global tasks, each entity is indexed with an identifier (ID) that usually comes from knowledge bases such as

Freebase. Exceptions are tasks in lexicon generation or population like entity set expansion (ESE) (Thelen and Riloff, 2002), which are global but without entity IDs. ESE usually starts from a few seed entities per set and completes the set using pattern-based methods.

Here, we address the task of *fine-grained name typing (FNT)*, a global prediction task, operating on the surface names of entities. FNT and ESE share applications in name lexicon population. FNT is different from ESE because we assume to have sufficient training instances for each type to train supervised models.

The challenging goal of FNT is to find the types of all entities a name can refer to. For example, "Washington" might refer to several entities which in turn may belong to multiple types, see Figure 1. In this example, "Washington" refers to "Washington DC (city)", "Washington (state)", or "George Washington (president)". Also, each entity can belong to several types, e.g., "George Washington" is a POLITICIAN, a PERSON and a SOLDIER, or "Washington (state)" is a STATE and a LOCATION.

Learning global representations for entities is very effective for global prediction tasks in IE (cf., Yaghoobzadeh and Schütze (2015)). For our task, FNT, we also learn a global representation for each name. By doing so, we see this task as a challenging evaluation for embedding models. We intend to use FNT to answer the following questions: (i) How well can embeddings represent distinctive information, i.e., different types or senses? (ii) Which properties are important for an embedding model to do well on this task?

We build a novel large-scale dataset of (name, types) from Freebase with millions of examples. The size of the dataset enables supervised approaches to work, an important requirement to be able to look at different subspaces of embeddings (Yaghoobzadeh and Schütze, 2016). Also, in FNT names are—in contrast to concept categorization datasets—multi-labeled, which requires to look at multiple subspaces of embeddings.

In summary, our contributions are (i) introducing a new evaluation method for word embeddings (ii) publishing a new dataset that is a good resource for evaluating word embeddings and is complementary to prior work: it is very large, contains more different classes than previous word categorization datasets, and allows the direct evaluation of embeddings without confounding factors like sentence context[1].

2 Related Work

Embedding evaluation. Baroni et al. (2014) evaluate embeddings on different intrinsic tests: similarity, analogy, synonym detection, categorization and selectional preference. Schnabel et al. (2015) introduce tasks with more fine-grained datasets. The concept categorization datasets used for embedding evaluation are mostly small (<500) (Baroni et al., 2014) and therefore measure the goodness of embeddings by the quality of their clustering. In contrast, we test embeddings in a classification setting and different subspaces of embeddings are analyzed. Extrinsic evaluations are also used (Li and Jurafsky, 2015; Köhn, 2015; Lai et al., 2015). In most tasks, embeddings are used in context/sentence representations with composition involved. In this work, we evaluate embeddings in isolation, on their ability to represent multiple senses.

Related tasks and datasets. Our proposed task is fine-grained name typing (FNT). A related task is entity set expansion (ESE): given a set of a few seed entities of a particular class, find other entities (Thelen and Riloff, 2002; Gupta and Manning, 2014). We can formulate FNT as ESE, however, there is a difference in the training data assumption. For our task, we assume to have enough instances for each type available, and, therefore, to be able to use a supervised learning approach. In contrast, for ESE, mostly only 3-5 seeds are given as training seeds for a set, which makes an evaluation like ours impossible.

Named entity recognition (NER) consists of recognizing and classifying mentions of entities locally in a particular context (Finkel et al., 2005). Recently, there has been increased interest in fine-grained typing of mentions (Ling and Weld, 2012; Yogatama et al., 2015; Ren et al., 2016; Shimaoka et al., 2016). One way of solving our task is to collect every mention of a name, use NER to predict the context-dependent types of mentions, and then take all predictions as the global types of the name. However, our focus in this paper is on how embedding models perform and propose this task as a good evaluation method. We leave the comparison to an NER-based approach for future work.

Corpus-level fine-grained entity typing is the

---

[1]Our dataset is available at: `https://github.com/yyaghoobzadeh/name_typing`

task of predicting all types of *entities* based on their mentions in a corpus (Yaghoobzadeh and Schütze, 2015; Yaghoobzadeh and Schütze, 2017; Yaghoobzadeh et al., 2018). This is similar to our task, FNT, but in FNT the goals is to find the corpus-level types of *names*. Corpus-level entity typing has also been used for embedding evaluation (Yaghoobzadeh and Schütze, 2016). However, they need an annotated corpus with entities. For FNT, however, pretrained word embeddings are sufficient for the evaluation.

Finally, there exists some previous work on FNT, e.g., Chesney et al. (2017). In contrast to us, they do not explicitly focus on the evaluation of embedding models, such that their dataset only contains a limited number of types. In contrast, we use 50 different types, making our dataset suitable for the type of evaluation intended.

3 Multi-label Classification of Word Embeddings

Word embeddings are global representations of word properties learned from the context distribution of words. Words are usually ambiguous and belong to multiple classes, e.g., multiple part-of-speech tags or multiple meanings. A good word embedding should represent all information about the word, including its multiple classes. Our evaluation methodology is based on this hypothesis and tries to test this through multi-label classification of word embeddings. Here, we focus on the semantic property of nouns and entity names. We try to find all categories or types of a noun given its embedding.

Multi-label classification of embedding has multiple advantages over current evaluation methods: (i) large datasets can be created without much human annotation; (ii) more fine-grained analysis of the results is possible through analyzing classification performance; (iii) it allows the direct evaluation of embeddings without confounding factors like sentence context.

4 Fine-grained Name Typing

We assume to have the following: a set of names N, a set of types T and a membership function $m : N \times T \mapsto \{0, 1\}$ such that $m(n, t) = 1$ iff name n has type t; and a large corpus C. In this problem setting, we address the task of *fine-grained name typing (FNT)*: we want to infer from the corpus for each pair of name n and type t whether $m(n, t) =$

1 holds.

For example, for the name "Hamilton", we should find all of the following: LOCATION, ORGANIZATION, PERSON, CITY, SPORTS_TEAM and SOLDIER, since "Hamilton" can describe entities belonging to those types. Another example is "Falcon" which is used for ANIMAL, AIRPLANE, SOFTWARE, ART. FNT sheds light on to which level these fine-grained types can be inferred from a corpus using embeddings.

4.1 Embedding-based Model

We aim to find $P(t|n)$, i.e., the probability of name n having type t. Given sufficient training instances for each type t, we can formulate the problem as a multi-label classification task. As input, we use a representation for n, learned from the corpus C. Distributional representations have shown to capture various types of information about a word, especially their categories or types (Yaghoobzadeh and Schütze, 2015).

After learning an embedding for n, we train two kinds of binary classifiers for each type t to to estimate $P(t|n)$: (i) linear: logistic regression (LR) with stochastic gradient decent; and (ii) non-linear: multi-layer perceptron (MLP) with one hidden layer and ReLU as the non-linearity. We use the Scikit-learn (Pedregosa et al., 2011) toolkit for training our classifiers.

5 Dataset

Using Freebase (Bollacker et al., 2008), we first retrieve the set of all entities E_n for each name n.[2] Then, we consider the types of all $e \in E_n$ the types of n. See Figure 1 for an example: all of the shown types belong to the name "Washington".

Since some of the about 1,500 Freebase types have very few instances, we map them first to the FIGER (Ling and Weld, 2012) type-set, which contains 113 types. We then further restrict our set to the top 50 most frequent types. See Table 5 for the list of types.

In order to be able to evaluate each embedding on its own, we divide our dataset into single-word (891,241 names) and multi-word (8,907,715 names). In this work, the multi-word set is not used. We then set a frequency threshold of 100 in our lowercased Wikipedia corpus [3] and select

---

[2] What we call "names" here are either *names* or *aliases* in the Freebase terminology.

[3] Our Wikipedia dump is from 2014.

3

/art, /art/film, /astral_body, /biology, /broadcast_network, /broadcast_program, /building, /building/restaurant, /chemistry, /computer/programming_language, /disease, /event, /food, /game, /geography/island, /geography/mountain, /god, /internet/website, /living_thing, /location, /location/body_of_water, /location/cemetery, /location/city, /location/county, /medicine/drug, /medicine/medical_treatment, /medicine/symptom, /music, /organization, /organization/airline, /organization/company, /organization/educational_institution, /organization/sports_team, /people/ethnicity, /person, /person/actor, /person/artist, /person/athlete, /person/author, /person/director, /person/engineer, /person/musician, /play, /product, /product/airplane, /product/instrument, /product/ship, /software, /title, /written_work

Table 1: List of the 50 types in our FNT dataset.

	#names	avg #types per name
train	50,000	3.78
dev	20,000	3.77
test	30,000	3.77

Table 2: Some statistics (number of names; average number of types per name) for our name typing dataset.

randomly 100,000 of our dataset names that pass this threshold. We then divide the names into train (50%), dev (20%) and test (30%). Some statistics of the single-word FNT dataset are shown in Table 2.

6 Experiments

6.1 FNT for Embedding Evaluation

Embedding models. We choose four different embedding models for our comparisons: (i) Skip-Gram (henceforth SKIP) (skipgram bag-of-words model) (Mikolov et al., 2013), (ii) CBOW (continuous bag-of-words model) (Mikolov et al., 2013), (iii) Structured SkipGram (henceforth SSKIP) (Ling et al., 2015), (iv) CWindow (henceforth CWIN) (continuous window model) (Ling et al., 2015), and SSKIP and CWIN are order-aware, i.e, they take the order of the context tokens into account, while SKIP and CBOW are bag-of-words models.

Results and analysis. We report the results for all embedding models using LR and MLP in Table 3. We use the following evaluation measures, which are used in entity typing (Yaghoobzadeh and Schütze, 2015): (i) ACC (accuracy): percentage of test examples where all predictions are correct, (ii) Micro-F1: the global F1 computed over all the predictions.

Models in lines 1-5 in Table 3 are trained on

	LR		MLP	
	ACC	Micro-F1	ACC	Micro-F1
1 CBOW	19.2	47.8	24.9	**54.6**
2 SKIP	22.6	49.3	**25.2**	53.5
3 CWIN	22.6	49.8	25.1	54.2
4 SSKIP	**23.4**	**50.5**	**25.2**	53.6

Table 3: Accuracy and micro-F1 results on FNT for different embedding models using two classifiers (LR and MLP). Best result in each column is bold.

the Wikipedia corpus. We set the min frequency in corpus to 100. Window size = 3; negative sampling with $n = 10$. Based on the results of LR, order-aware architectures are better than their bag-of-words counterparts, i.e., SSKIP > SKIP and CWIN > CBOW. Overall, SSKIP is the best using LR classification. In MLP results, however, CBOW works best on micro-F1 measure and SSKIP and SKIP are bests on accuracy. There is no significant difference between CBOW and CWIN, or SSKIP and SKIP, respectively. Overall, the nonlinear classifier (MLP) with one hidden layer outperforms the linear classifier (LR) substantially, emphasizing that the encoded information about different types is easier to extract with stronger models.

Analysis on the number of name types. As a separate analysis, we measure how the classification performance depends on the N number of types of a name. To do so, we group test names based on their number of types. We keep the groups that have more than 100 members. Then, we plot the F1 results of CBOW and CWIN models trained using MLP classifier in Figure 2.

As it is shown, both models get their best results on names with $N = 2$. We suppose that the bad performance of $N = 1$ is related to the fact that one-type names have missing types in our dataset due to the incompleteness of Freebase. The worse F1 of $N >= 3$ compared to $N = 2$ is expected since bigger N means that the models need to predict more types from the name embeddings. From $N = 4$, somewhat surprisingly the F1 increases as N increases. This is perhaps related to the frequency of names in the corpus, and its relation to the number of names types: as N increases, the frequency of words increases and the embedding has a better quality. However, this is only a hypothesis and more investigation is required. The other observation is in the trend of CBOW and CWIN results. CBOW is worse for $N <= 2$, but
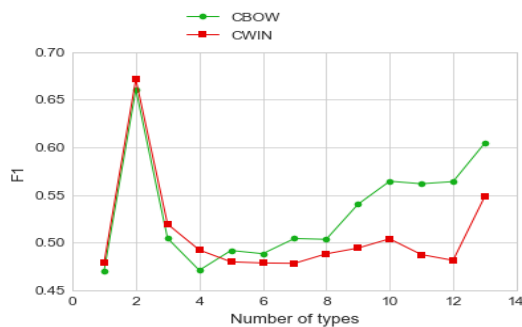
4

Figure 2: Micro-F1 for names with different number of types.

it works clearly better for $N > 2$. This shows that the embedding models behave differently for different number of classes they belong to. This could also be related to the frequency of words. Analysis of the reasons would be interesting. We leave it for the future work.

7   Conclusion

We proposed multi-label classification of word embeddings using the task of fine-grained typing of entity names. The dataset we built is a resource that is complementary to prior work in embedding evaluation: it is very large, its examples are multi-labeled with very fine-grained classes, and it allows the direct evaluation of embeddings without the need for context. We analyzed the performance of different embedding models on this dataset, showing differences in their performance as well as some of their limits in representing types accurately and completely.

More analysis and evaluation is necessary though, but we believe by using this kind of dataset, we are able to do much more than what we could do before with the small manually built word similarity and categorization benchmarks.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pages 238–247.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250.

Sophie Chesney, Guillaume Jacquet, Ralf Steinberger, and Jakub Piskorski. 2017. Multi-word entity classification in a highly multilingual environment. In *MWE*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan. Association for Computational Linguistics.

Sonal Gupta and Christopher D. Manning. 2014. Improved pattern learning for bootstrapped entity extraction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*, pages 98–108.

Arne Köhn. 2015. What?s in an embedding? analyzing word embeddings through multilingual evaluation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Lisbon, Portugal.

Siwei Lai, Kang Liu, Liheng Xu, and Jun Zhao. 2015. How to generate a good word embedding? *CoRR*, abs/1507.05523.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal.

Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1299–1304.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, Toronto, Ontario, Canada.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1378, Austin, Texas. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia. Association for Computational Linguistics.

Dana Rubinstein, Effi Levi, Roy Schwartz, and Ari Rappoport. 2015. How well do distributional models capture different types of semantic knowledge? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 726–730.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. An attentive neural architecture for fine-grained entity type classification. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 69–74, San Diego, CA. Association for Computational Linguistics.

Michael Thelen and Ellen Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 214–221. Association for Computational Linguistics.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schuetze. 2018. Corpus-level fine-grained entity typing. *Journal of Artificial Intelligence Research*, 61:835–862.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2015. Corpus-level fine-grained entity typing using contextual information. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 715–725, Lisbon, Portugal.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2016. Intrinsic subspace evaluation of word embedding representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 236–246, Berlin, Germany. Association for Computational Linguistics.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2017. Multi-level representations for fine-grained typing of knowledge base entities. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 578–589. Association for Computational Linguistics.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 291–296, Beijing, China. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 2335–2344.

6