Using word2vec to Build a Simple Ontology Learning System

Gerhard Wohlgenannt, Filip Minic

Vienna Univ. of Economics and Business, Welthandelsplatz 1, 1200 Wien, Austria
{gerhard.wohlgenannt,filip.minic}@wu.ac.at
http://www.wu.ac.at

Abstract. Ontology learning has been an important research area in the Semantic Web field in the last 20 years. Ontology learning systems generate domain models from data (typically text) using a combination of sophisticated methods. In this poster, we study the use of Google's word2vec to emulate a simple ontology learning system, and compare the results to an existing "traditional" ontology learning system.

Keywords: ontology learning, word2vec, term extraction

1 Introduction

Ontologies are the vocabulary used on the Semantic Web. Manual ontology construction is an expensive effort, therefore a number of systems to automatically extract ontologies from data (often natural language text) have been proposed. Those systems bootstrap the ontology construction process by providing ontology engineers with learned ontologies. Ontology learning (OL) system are usually big and complex frameworks that use different data sources and techniques to extract terms, synonyms, concepts, taxonomies, etc. from data.

Mikolov et al. [3] present a system called *word2vec*, which, despite its simplicity, has been shown to be very effective to provide similar terms, but also to extract certain syntactic and semantic relations simply using vector operations such as addition and subtraction of vectors.

The complexity of traditional OL systems makes them hard to use, maintain and extend. Replacing or improving some complex parts of traditional OL systems would therefore benefit the community. This leads us to the following research questions: How well is the state-of-the-art tool word2vec suited to execute OL tasks such as term and taxonomy extraction from text? Which differences regarding results are to expected?

2 Related Work

Ontology learning (OL) has been an active research field since around year 2000, many state-of-the-art systems like OntoLearn Reloaded [4] use a plethora of methods on domain text to extract terms, concepts, taxonomic relations and

finally construct a taxonomy. Text2Onto [1] uses many algorithms from the NLP and Information Retrieval fields to generate ontology models. Manzano-Macho et al. [2] emphasize the benefits of using heterogeneous sources of evidence in OL, in order to leverage redundant information in various sources. Most OL systems are geared to generate lightweight ontologies, it is unclear if fully automated learning of heavyweight ontologies is feasible at all [6].

Word2vec [3] computes continuous vector representations for large text data sets. Word2vec outperforms the state-of-the-art in word similarity tasks, and provides high performance for measuring syntactic and semantic similarities.

In this publication we present our initial results from integrating word2vec into traditional OL.

3 Methods

In this section we briefly introduce the two methods that will be compared, the OL framework developed at WU Vienna, and the word2vec implementation.

Our Ontology Learning system: Due to limitations in space, we can only give a brief overview of the workflow of our OL system, more details are found eg. in Wohlgenannt [5]. The input to our OL system are heterogeneous evidence sources: domain text, Wordnet, DBpedia, and APIs of some social media sites. From those sources the system extracts evidence for important terms and relations between terms. The system learns domain ontologies from scratch using a small amount (for example: two) seed concepts in that domain. The process is a follows (simplified): (i) Extract new term and relation candidates from evidence sources based on seed concepts with various methods: co-occurrence, Hearst patterns, etc. (ii) Integrate all evidences into a big semantic network. (iii) Use spreading activation (a neural network method) to find the 25 most important concept candidates. (iv) Evaluate the 25 candidates with crowdsourcing or domain experts for domain relevance. (v) Position the selected new concepts in the ontology, which gives an extended ontology. (vi) Use the extended ontology as new seed ontology and go back to step i). We typically run this iterative ontology extension cycle for 3 times and then halt.

word2vec: Word2vec is a two-layer neural net, which uses natural language text as input. The output are continuous feature vectors of a given size (eg. 300 dimensions) for the input words (or phrases). Word2vec trains neural nets to reconstruct the linguistic contexts of words, using two methods: continuous bag-of-words (CBOW) or continuous skip-gram. With CBOW, the model predicts the current word using a window of surrounding words. Word2vec is well suited to provide high-quality *similar terms* for an input term, and also allows vector operations. A well-know example of this is: `king` is to `queen` what `man` is to `x`. And the vector operation $king - queen + man$ should than provide *woman* as best guess. With word2vec, we use exactly the same workflow as in our OL system: Three iterations learning 25 terms each, and using the confirmed terms as new seed terms in the next iteration.

4 Evaluation

4.1 Evaluation Setup

We compare the results for three methods. First, *word2vec – unigrams* is a word2vec model trained on single words in the corpus. *word2vec – bigrams* uses single words and bigrams from the corpus, and finally, *OL* uses the results from our OL system. With these three methods, we extract concept candidates from four corpora each. The corpora mainly consist of news media coverage about the *climate change* domain mirrored in a specific month. The basic setup is always the same, we start from a seed ontology, and do three extension steps, where we collect 25 concept candidates, and add the candidates which have been manually judged as domain-relevant to the ontology.

4.2 Results

Table 1 presents the percentages of relevant concept candidates according to a manual evaluation by domain experts. The experiments suggest that *word2vec* is very well suited for the term and concept extraction step in OL.

Data from Period:	**word2vec – unigrams**	**word2vec – bigrams**	**OL**
May 2015	78.7%	81.3%	68.0%
June 2015	81.3%	75.0%	73.3%
July 2015	82.7%	77.3%	69.0%
August 2015	84.0%	85.3%	62.0%
Average	81.7%	79.8%	68.1%

Table 1. Percentage of domain-relevant new concept candidates collected with 3 methods for 4 underlying corpora.

We keep the process for *word2vec* as simple as possible. After word2vec model generation with the standard word2vec scripts based on the plain-text corpus, we apply the built-in *word2vec* similarity function to get terms related to the seed terms. On the plus side, the word2vec implementation is extremely simple[1], and provides a high-percentage of relevant concept candidates. On the minus side, candidates suggested by word2vec are (as expected) sometimes even too strongly related to the seed terms, for example: syntactic variations such as plural forms or near-synonyms. In future work, we will address the issue of *too similar* terms/concepts suggested by word2vec with i) detection and filtering of syntactic variations and synonyms, ii) using vector operations to detect interesting relations as well as new concept candidates. One interesting finding was, that with our traditional OL system, the more concepts already exist in the ontology, the lower the quality of generated new candidates – but with the word2vec system is was the other way around.

[1] Source-code and data found at: `https://aic.ai.wu.ac.at/~wohlg/iswc2016`

In addition to concept detection, we evaluated word2vec for taxonomy building. For this task, we first collected some predefined term pairs with taxonomic relations in the sense of *skos:broader* such as *tree / forest* or *methane / greenhouse gas*. We then applied those taxonomic term pairs with the word2vec analogy function, for example *tree is to forest* what *coal is to X* – and let word2vec generate suggestions for *X*. If *X* was a concept existing in the ontology, we manually evaluated the correctness of the taxonomic relation. For the ontologies generated with method (i) *word2vec – unigrams*, the word2vec-based taxonomy generation method suggested 64 taxonomic relations, of which 34 were evaluated as correct (53.13%). In the *word2vec – bigrams* method we had 209 suggestions, and 101 correct (48.33%). An accuracy of around 50% on taxonomic relation suggestion is not very impressive, but we see lot of room for improving the system by parameter settings and using bigger corpora in future work. A source of error were eg. wrong directions in taxonomic relations suggested by word2vec.

5 Conclusions

In this poster we presented first results from emulating ontology learning (OL) tasks by using word2vec. Results show that word2vec can be very useful in term and concept extraction, for learning taxonomic relations more work needs to be done. The contributions are as follows: (i) finding and implementing simple substitutes of some complex parts of OL systems, and (ii) evaluating word2vec for term/concept extraction and taxonomy generation. Future work will include large scale evaluations in various domains and including tools for synonym detection. Furthermore, we will improve the extraction of taxonomic relations by parameter tuning, the use of bigger corpora, and more training relations. Finally, we also plan to see how well word2vec works for the OL task of detecting non-taxonomic relations, and compare it to existing approaches.

References

1. Cimiano, P., Völker, J.: Text2onto: A framework for ontology learning and data-driven change discovery. In: Proceedings of the 10th Int. Conf. on NLP and IS. pp. 227–238. NLDB'05, Springer-Verlag, Berlin, Heidelberg (2005)
2. David Manzano-Macho, A.G.P., Borrajo, D.: Unsupervised and domain independent ontology learning: Combining heterogeneous sources of evidence. In: Calzolari, e.a. (ed.) Proceedings of LREC'08. ELRA, Marrakech, Morocco (May 2008)
3. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
4. Velardi, P., Faralli, S., Navigli, R.: OntoLearn Reloaded: A Graph-based Algorithm for Taxonomy Induction. Computational Linguistics 39(3), 665–707 (2013)
5. Wohlgenannt, G.: Leveraging and balancing heterogeneous sources of evidence in ontology learning. In: et al., G. (ed.) ESWC 2015, Portoroz, Slovenia. Lecture Notes in Computer Science (LNCS), vol. 9088, pp. 54–68. Springer (2015)
6. Wong, W., Liu, W., Bennamoun, M.: Ontology learning from text: A look back and into the future. ACM Computing Surveys 44(4), 20:1–20:36 (Sep 2012)