Verb Sense Induction using Phrase Embedding Compositions

Dan Prendergast

Daniel.Prendergast@colorado.edu

Sam Young

Samuel.A.Young@colorado.edu

Abstract

In this paper, we experimented with different methods of composing the word vector of a verb with that of its object. We found that by using simple vector composition methods and k-means clustering it is possible to achieve significant results in terms of aligning verb meanings to PropBank gold standard verb senses.

I. Introduction

One of the quintessential problems of computational linguistics is the ambiguity of words in a language. Consider the word “run.” In some contexts, “run” can mean movement under an agent’s own direction, as in “to run a marathon.” In another context, the word can mean to exercise leadership over a group or entity, as in “to run a company.” This type of ambiguity makes automated processing of text and speech difficult. Without the ability to make the distinction between senses, an automated assistant may mistake your intent to “run a marathon this March” as an intent to plan and coordinate a sporting event rather than mere participation in said event.

II. Problem

The task of identifying the proper meaning of different uses of words is called word sense disambiguation (WSD). WSD can largely be divided into supervised learning, unsupervised learning, and knowledge-based approaches (Navigli, 2009). The knowledge-based approaches, such as WordNet (Miller, 1995), are often used as a gold standard for evaluating other WSD methods. The disadvantage of supervised learning and knowledge-based approaches is the considerable effort required to create manually annotated datasets. On the other hand, unsupervised methods do not require labeled data. The disadvantage of unsupervised learning is that they cannot assign sense labels. Instead they perform word sense induction (or discrimination). That is, they can identify when occurrences of a word fall into different sense classes, but they cannot apply semantic labels to the classes.

Klapaftis and Manandhar (2013) provide an overview of word sense induction (WSI) methods. These methods can be categorized as Bayesian, graph-based, or vector-based approaches. Bayesian approaches such as Brody and Lapata (2009) and Blei et al. (2003) use a generative approach to WSI using probability distributions of words and senses. The graph-based methods (Dorow and Widdows, 2003; Véronis, 2004; Agirre et al., 2006b) first construct a graph with context words and target words as vertices and co-occurrences of those words as edges. After the corpus is ingested and the graph built, graph reduction algorithms are used to find the significant clusters of vertices (Klapaftis and Manandhar, 2013). Lastly, vector-based WSI uses a co-occurrence matrix to develop a vector representation for a target word. Various clustering techniques can then be applied to the vectors to induce the separate word senses as in Pedersen (2007), Niu et al. (2007), and Pinto et al. (2007).

III. Approach

For the purposes of this paper, we intend to induce verb sense distinctions using verb-object phrase embeddings. Our approach is based on the assumption that the object of a verb adds context that assists in disambiguating the verb sense. Dligach and Palmer (2008) demonstrate that by utilizing the semantic context of verb objects using a dependency-parsed corpus, error rates of verb disambiguation tasks can be decreased by as much as 15%.

One method of creating phrase embeddings is through simple composition of individual word embeddings. Mitchell and Lapata (2008) formalize additive and multiplicative composition models. Their work shows that compositions involving multiplication of subject and verb embeddings performed better than simple addition on a sentence similarity task (Mitchell and Lapata, 2008).

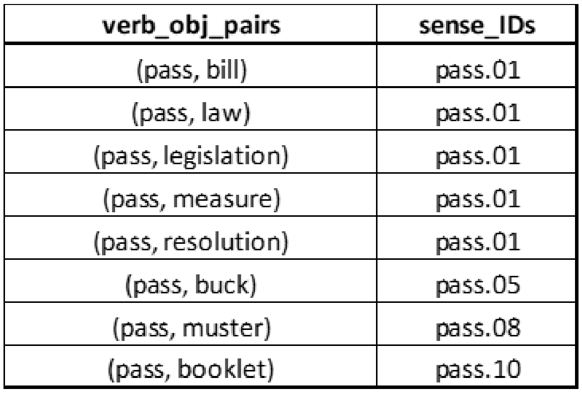
In this project, we will create phrase embeddings through composition of verb and object word embeddings. Then we will evaluate the effectiveness of discriminating different verb senses by clustering these phrase embeddings. We hypothesize that verb senses can be effectively induced using verb-object phrase embeddings. The advantage of this approach is that it can be performed locally on a sentence-by-sentence basis, unlike the graph-based or Bayesian WSI approaches described above. Such a system could be useful in speech recognition or translation systems where disambiguation must be performed in real-time.

The basic steps of our approach in this research include:

- Extraction of Verb-object pairs from PropBank
- Calculation of phrase embeddings
- Clustering of phrase embeddings
- Alignment of senses to clusters

Extraction of Verb-Object Pairs

We chose 28 verbs from the University of Colorado's Unified Verb Index based on the perceived number of meanings, frequency of usage, and number of Propbank senses. Then, using the NLTK labeled treebank/propbank dataset, we extracted sentences containing these verbs as predicates with their Arg1 roles. Although Arg1 does not strictly denote the grammatical object of a verb, it is computationally cheaper to extract semantic role labels than it is to load and interpret syntax trees to extract the syntactically correct direct object, and for the purposes of this experiment it made sense to focus primarily on semantic rather than syntactic criteria. From the Arg1 phrase we manually coded a single noun that semantically represented the direct object of the predicate.



verb_obj_pairs	sense_IDs
(pass, bill)	pass.01
(pass, law)	pass.01
(pass, legislation)	pass.01
(pass, measure)	pass.01
(pass, resolution)	pass.01
(pass, buck)	pass.05
(pass, muster)	pass.08
(pass, booklet)	pass.10

Table 1: Verb-object pairs extracted from PropBank

From the lemmatizations of the sentence predicates and direct objects we formed a list of verb-object pairs. We also collected the Propbank verb sense for each verb-object pair to be used later during cluster alignment. Any time a phrase contained a verb with multiple Arg1s the verb was split into multiple verb-object pairs. We then deleted any duplicate verb-object-sense tuples from the list. This process yielded 219 total samples (i.e., verb-object pairs). Within this set were 28 unique verbs, and the average number of unique objects per verb was 7.8. See Table 1 above for a sample of the extracted data.

Pre-trained Word Embeddings

For this task we decided to test our method using pre-trained word vectors from two of the most widely used word embedding models: word2vec and GloVe. These models work by counting the frequency with which words are used within a certain distance of each other in context. Co-occurrence probabilities are then used to train the vectors either by predicting a word using the surrounding context or predicting the surrounding context based on an individual word. The methods used to train word2vec and GloVe vectors are explained in further detail in Mandelbaum and Shalev (2016) and Pennington et al. (2014).

We used the 300 dimension wiki2vec model trained on the Google News text corpus and compared it to the 300 dimension GloVe model trained on the Wikipedia text corpus.

Calculation of Phrase Embeddings

The phrasal composition methods we used for this study are based on those outlined by Mitchell and Lapata (2008). The paper used 7 methods of composition. Three of these models were additive, meaning that the verb and object embeddings were added together element wise to create a combined phrase vector. We decided only to use the simplest of the addition methods for expediency. The additive calculation is given by…

$$p_i = u_i + v_i$$

where p is the phrase embedding, u is the object embedding and v is the verb embedding. The subscript i indicates the i^{th} element of the embedding. The overall performance of the different methods of addition was poor relative to the multiplication and combined methods. Multiplication involves simple element-wise multiplication between the verb and object embeddings as given by…

$$p_i = u_i * v_i \,.$$

Lastly, the combined method involves adding the results of both the addition and multiplication vectors together into a single embedding as shown in…

$$p_i = u_i + v_i + (u_i * v_i)\,.$$

Also, it is important to note that vectors were not normalized after the compositions were calculated.

For each verb-object pair extracted from the PropBank dataset, we calculated a phrase embedding using one of the three compositions defined above. This process was completed twice -- once using the GloVe dataset as the source for verb and object embeddings and again using the word2vec dataset. This resulted in 6 complete sets of phrase embeddings for the set of verb-object pairs:

1. GloVe - Addition
2. GloVe - Multiplication
3. GloVe - Combination
4. word2vec - Addition
5. word2vec - Multiplication
6. word2vec - Combination

Clustering of Phrase Embeddings

The next step in our analysis was to perform clustering of the verb-object pairs using their phrase embeddings for each verb. This was performed using k-means clustering, where k was the number of senses present in the Propbank data for that verb. For example referring to Table 1, we see that the PropBank dataset demonstrated 4 senses of the verb “pass.” Therefore the phrase embeddings for "pass" were clustered into 4 groups. At this point the clusters were assigned arbitrary numbers (e.g., "Cluster 0", "Cluster 1", etc.) to aid in alignment with senses in the next step.

Alignment of Senses to Clusters

Our approach has no inherent way of determining which sense is most correctly applied to each of the verb-object clusters. Therefore, we must use the PropBank data as a standard by which to find the best alignment of senses to clusters.

An alignment is simply an assignment of each of the verb's senses from the PropBank data to one of the cluster numbers found above. One example alignment is the following:

cluster 0 = sense_ID pass.01
cluster 1 = sense_ID pass.05
cluster 2 = sense_ID pass.08
cluster 3 = sense_ID pass.10

With an alignment, we can assign a predicted sense to each verb-object pair for a given verb and verify its accuracy against the sense given by the ProbBank data using the following calculation...

$$Accuracy\ =\ Count(s_p = s_T)/N$$

where s_p is the predicted sense of the verb-object pair, s_T is the true sense provided by PropBank, and N is the number of verb-object pairs for that verb. We perform this calculation for every permutation of possible alignments to find the best one for each verb in the complete set of verb-object pairs.

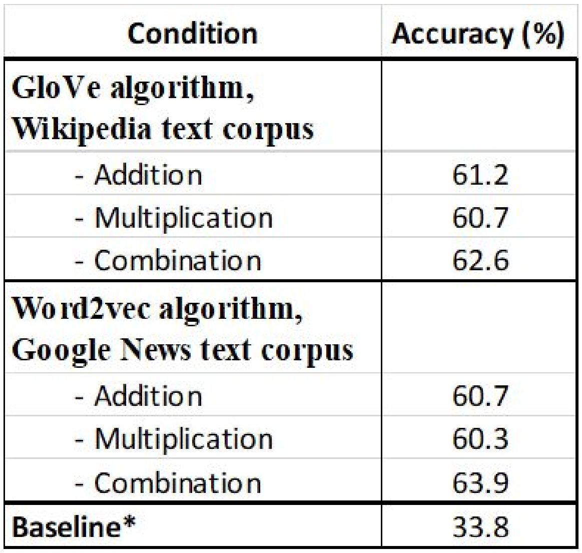
IV. Evaluation

The intent of this project is to evaluate which word embedding set (GloVe vs. word2vec) and which phrase embedding composition (addition, multiplication, vs. combination) produces the best induction of verb senses. This evaluation requires a 2 x 3 factorial design with two variables -- word embedding set (2 levels) and embedding composition (3 levels). The measure we will use for this evaluation is accuracy of sense prediction, defined in the equation above.

We begin by calculating an overall accuracy for each of the six sets of verb-object pairs and phrase embeddings defined above (GloVe - Addition, GloVe - Muliplication, etc.). As a baseline, we will compare this to the accuracy we could expect to achieve from random assignment of the verb-object pair to one of the senses found in the PropBank dataset for that pair's verb.

V. Results and Discussion

In both datasets, addition outperformed multiplication while a combination of both techniques yielded the best results (see Table 2 below). This contradicts the results achieved by Mitchell and Lapata (2008), whose study demonstrated both multiplication and combination by far outperforming the results of any of the addition methods.



Condition	Accuracy (%)
GloVe algorithm, Wikipedia text corpus	
- Addition	61.2
- Multiplication	60.7
- Combination	62.6
Word2vec algorithm, Google News text corpus	
- Addition	60.7
- Multiplication	60.3
- Combination	63.9
Baseline*	33.8

Table 2: VSI accuracy as a function of word embedding sets and composition methods

The differences in our results may be due to any number of factors. The relationship between a verb and its object may be computationally distinct from that of its subject. This interpretation would suggest that different types of vector composition methods may yield different results for different kinds of tasks, in which case further experimentation with composition methods is warranted. Another possibility is that the difference in goals between the two studies, namely the usage of human labeling versus semantic role labeling, caused a major difference in the results. There may be flaws endemic to both methods of similarity evaluation techniques. Human similarity comparisons may not accurately represent the logic that computers use or should use to disambiguate meaning. Contrastively, Propbank roles may be flawed in how they represent meaning and may not distinguish between all of the different flavors of meaning endemic to a specific word. Both methods of evaluation are inherently subjective. It may be warranted to experiment with different kinds of verb sense evaluation methods in tandem with the argument role composition, i.e. evaluating subject/verb pairs with Propbank roles and evaluating verb/object pairs with human similarity ratings.

References

Agirre, E., Martı´nez, D., Lo´pez de Lacalle, O., & Soroa, A. (2006b). Two graph-based algorithms for state-of-the-art wsd. In Proceedings of the conference on empirical methods in natural language processing (pp. 585–593). Sydney, Australia: ACL.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. J. Mach. Learn. Res., 3, 993–1022.

Brody, S., & Lapata, M. (2009). Bayesian word sense induction. In Proceedings of the 12th conference of the european chapter of the association for computational linguistics, EACL ’09 (pp. 103–111). Stroudsburg, PA, USA: Association for Computational Linguistics.

Dligach, Dmitriy, and Martha Palmer. "Novel semantic features for verb sense disambiguation." *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics, 2008.

Dorow, B., & Widdows, D. (2003). Discovering corpus-specific word senses. In Proceedings of the 10th conference of the European chapter of the ACL (pp. 79–82). Budapest, Hungary: ACL.

Kim, Joo-Kyung, Marie-Catherine de Marneffe, and Eric Fosler-Lussier. "Neural word embeddings with multiplicative feature interactions for tensor-based compositions." *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. 2015.

Kingsbury, Paul, and Martha Palmer. "From TreeBank to PropBank." *LREC*. 2002.

Klapaftis, Ioannis P., and Suresh Manandhar. "Evaluating word sense induction and disambiguation methods." *Language resources and evaluation* 47.3 (2013): 579-605.

Mandelbaum, Amit, and Adi Shalev. "Word embeddings and their use in sentence classification tasks." *arXiv preprint arXiv:1610.08229* (2016).

Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

Miller, George A. "WordNet: a lexical database for English." *Communications of the ACM* 38.11 (1995): 39-41.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In Association for Computational Linguistics (ACL), pages 236–244.

Navigli, Roberto. "Word sense disambiguation: A survey." *ACM computing surveys (CSUR)* 41.2 (2009): 10.

Niu, Z.-Y., Ji, D.-H., & Tan, C.-L. (2007). I2r: Three systems for word sense discrimination, chinese word sense disambiguation, and english word sense disambiguation. In Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007) (pp. 177–182). Prague, Czech Republic: Association for Computational Linguistics.

Pedersen, T. (2007). Umnd2: Senseclusters applied to the sense induction task of senseval-4. In Proceedings of the fourth international workshop on semantic evaluations (pp. 394–397). Prague, Czech Republic: ACL.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

Pinto, D., Rosso, P., & Jime´nez-Salazar, H. (2007). Upv-si: Word sense induction using self term expansion. In Proceedings of the fourth international workshop on semantic evaluations (SemEval2007) (pp. 430–433). Prague, Czech Republic: Association for Computational Linguistics.

Véronis, J. (2004). Hyperlex: lexical cartography for information retrieval. Computer Speech & Language, 18(3), 223–252.