Concept Extensions as the Basis for Vector-Space Semantics: Combining Distributional and Ontological Information about Entities

Jackie Chi Kit Cheung
McGill University
3480 University
Montreal, Quebec, Canada
jcheung@cs.mcgill.ca

Abstract

We propose to base the development of vector-space models of semantics on concept extensions, which defines concepts to be sets of entities. We investigate two sources of knowledge about entities that could be relevant: distributional information provided by word or phrase embeddings, and ontological information derived from a knowledge base. We develop a feedforward neural network architecture to learn entity representations that are used to predict their concept memberships, and show that the two sources of information are actually complementary. In an entity ranking experiment, the combination approach that uses both types of information outperforms models that only rely on one of the two. We also perform an analysis of the output using fuzzy logic techniques to demonstrate the potential of learning concept extensions for supporting inference involving classical semantic operations.

1 Introduction

The *extensional definition*, or *denotation*, of a concept is the set of entities in the world to which that concept applies. For example, the definition of *Celebrity* would be the set of entities in the world, including *Will Smith*, *Paris Hilton*, etc.

In formal semantics and pragmatics, this conception of meaning has played an important role in the accounts of a wide range of compositional constructions, including definite and indefinite articles, quantifiers, presuppositions, and intersective adjectives. For example, the extension of a noun phrase such as *"red apple"* that is composed of a noun and a modifying adjective is derived by taking the set intersection of the extensions of *"red"* and *"apple"*. In an applied setting, explicitly enumerating the members of these extensions seems to be an impossible task, as there are large numbers of entities and relations, not to mention infinitely many possible contexts and domains. Thus, the direct application of this view of semantics would seem to be confined to limited domains.

Distributional semantics is a potential solution to this problem. The long-touted advantages of distributional approaches are that they can be easily trained from a large corpus, and they enable a graded notion of similarity. Typically, such models are trained to optimize distributional criteria based on similarity correlations or predicting a word in context. However, it is not enough to rely solely on these criteria. Similarity only supports *relative* reasoning about relations between concepts, and it is difficult to adapt such measures to make *absolute* inferences about the truth value of a proposition. The applications of distributional semantics (DS) to date have reflected this bias. The most common approach to evaluate DS models has been to correlate predicted similarity judgments against judgments gathered from humans (Finkelstein et al., 2002; Agirre et al., 2012). More recent applications in paraphrase detection (Socher et al., 2011), textual entailment (Beltagy et al., 2013) and analogical reasoning (Mikolov et al., 2013) are also primarily concerned with the relationships between phrases.

A more serious issue is that distributional semantics alone seems to be insufficient for handling rarely occurring events and entities, if we treat them as just another target phrase in the corpus. Consider the following passage:

Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality (CVSC), pages 22–31,
Beijing, China, July 26-31, 2015.

(1) *He is an American actor, producer, and rapper. As of 2014, 17 of the 21 films in which he has had leading roles have accumulated worldwide gross earnings of over $100 million each.*

Given just this short description of the entity, we are able to make several inferences about its properties. For example, we are able to infer that this entity is a male human, working in the entertainment industry. He can most likely vote in American elections, obtain a passport, and he is likely a wealthy celebrity, given the success of the movies he has acted in. We might even be able to guess the identity of this person (Will Smith)[1].

While it may be possible to learn these characteristics from the contexts of the bigram *"Will Smith"* in a large training corpus, this is less plausible for a rarely occurring, or perhaps an entirely invented entity. Clearly, these inferences are enabled by extracting the concept and relational information present in the local context, then relating them to other concepts of interest based on our knowledge of the world.

In this paper, we propose to use concept extension predictions as the overall training objective of a vector-space model of semantics. While distributional information will still be a crucial component of our model, what distinguishes our approach is that it optimizes directly for an objective which is well justified by compositional theories of semantics, rather than an objective that is internal to considerations within distributional semantics such as similarity measurements.

To predict these concept extensions, we train a model that learns a representation of an input entity using features derived from distributional semantics and ontological information derived from a knowledge base. Our model, which we call *Ontological Distributional Semantics*, employs a simple feedforward neural network architecture to learn interactions between these two sources of information.

We conduct experiments on Freebase (Bollacker et al., 2008), taking Freebase types to be concepts, and the entity set that the Freebase type contains to be that concept's extension. The results of an entity ranking experiment show that Ontological Distributed Semantics outperforms either distributed representations or ontological information alone across three entity classes.

[1]This passage is an edited version of his Wikipedia article.

Because a large, complete knowledge base may not always be available, we further test our model under conditions in which there is incomplete ontological knowledge about an entity, and we analyze the relative contributions of the distributional and ontological components of our model.

Finally, to illustrate how our approach can take advantage of insights from classical approaches to semantics, we develop a method to extract semantic relations between concepts from the output predictions of our model without further training using fuzzy set logic operations. These results argue for the importance of learning concept extensions not just to develop a better model of entities, but also as a potential method to better integrate distributional semantics with formal, compositional approaches to semantics.

2 Related Work

Several models have recently been proposed which combine distributional with ontological information (Fried and Duh, 2014; Yang et al., 2014). The goal of these papers is to encode the ontological relationships as some kind of regularity in the learned vector space, usually as a linear transformation; e.g., that objective encourages there to be a consistent vector addition operation that represents the part-of relationship between two concepts. By contrast, our work argues for an entirely different kind of objective function for a vector-space model motivated by classical compositional semantics.

Herbelot and Ganesalingam (2013) investigate KL-divergence of a semantic vector as a simple information-theoretic measure to determine hypernym/hyponym relations, but found that this was outperformed by a word frequency baseline. Other work employs distributional similarity to learn to cluster concepts into a hierarchy (Yamada et al., 2009, for example). There have also been supervised methods for hypernymy detection (Roller et al., 2014, for example). Typically, this is done for upward-entailing concept-to-concept reasoning, for example between word pairs (e.g., *van* entails *car*) as in the BLESS data set (Baroni and Lenci, 2011).

Another thread of related work is in relation extraction (Banko et al., 2007; Bunescu and Mooney, 2007; Riedel et al., 2013, for example), and knowledge base population, such as the TAC shared task (McNamee and Dang, 2009). This

work is concerned with extracting the relationships between entities, in order to improve the quality of a database. Our work can be seen as a way of integrating distributional semantics into large-scale reasoning about entities.

Most recently, Gupta et al. (2015) investigate a similar problem, using a logistic regression model to map features derived from distributional methods to referential properties of countries that are derived from Freebase. In our work, we explore the effect of combining distributional and ontological information, and perform a number of analyses on the output of our models.

3 Framework and Model

Our model is designed to learn entity representations that are useful for predicting concept extensions, which are sets of entities in the domain. Let $C = \{c_1, c_2, ...\}$ be the set of concepts of interest, and $E = \{e_1, e_2, ...\}$ be the set of entities. Since we are interested in extensional meaning, each concept c is defined by its extension, *exten*(c), a set of elements drawn from E. Rather than explicitly enumerating these sets, we instead aim to learn a function $f : E \rightarrow \mathcal{P}(C)$ that maps an input entity to the concepts of which it is an element. For example, f(*Will Smith*) would evaluate to the concepts *Male* and *Actor*, but also ¬*Female*, among others.

We frame this as a supervised multi-label classification problem. For an entity $e \in E$, the input to the classifier is a feature vector representation of the entity, $\vec{x}$. The classifier predicts a binary output vector $\vec{y}$ of length $|C|$, in which $y_k = 1$ means that $e \in$ *exten*(c_k), and $y_k = 0$ means that $e \notin$ *exten*(c_k). In our experiments, we will actually assume that the classifier makes probabilistic, "soft" decisions, so that the entries of the output vector will range from 0 to 1, representing the predicted probability of the entity being a member of the concept extension.

It is possible to view this task as a series of standard binary classification problems, one for each of the concepts. However, this would require training a large number of concept-specific models. Our hope in learning entity representations is that they will be more generally useful, for example, in a compositional setting in which inferences are to be made about phrases containing entities for which we have already learned a representation.

3.1 Input features

We now specify the input feature vector representation of the entity, as well as a learning algorithm for the function f. Our full model combines ontological information with pre-trained distributional semantic vectors to learn the extensional meaning of concepts. To measure the effect of each of these components, we also train baseline versions of the model that omit one or the other feature set. Thus, we compare the following three sets of features:

DS We derive a *distributional vector* of features from word2vec, a popular recent approach to distributional semantics (Mikolov et al., 2013). We use the 300-dimensional pre-trained vectors available on their website, which include both single-word and multi-word entities. We chose word2vec as it is a popular recent model of distributional semantics which has been shown to work well on a variety of existing semantic tasks (Baroni et al., 2014). We leave the comparison of this model to other recent distributional semantic models (Pennington et al., 2014, for example) to future work.

ONTO For the ontological features, we derive an *ontological vector* of an entity from its Freebase entry. Each dimension of the ontological vector corresponds to a concept, represented by a Freebase type. The vector takes a value of 1 at that dimension if the entity is an instance of that concept, and 0 otherwise. For example, if the first three dimensions of the ontological vector correspond to the concepts *Male*, *Actor*, and *Female*, their values for the ontological vector of *Will Smith* would be 1, 1, and 0, respectively.

ONTODS We concatenate the above two feature vectors into an ontological distributional semantic vector.

3.2 Learning algorithm

The learning algorithm of our model is a simple feedforward neural network. The neural network has one hidden layer, the entity representation, which is then used to predict the output vector $\vec{y}$. The network is trained by stochastic gradient descent with a mean squared error loss, a sigmoid nonlinearity and weight decay. All of the parameters to the model are tuned according to performance on a held-out development set (Section 4.1).

Using a neural network offers several advantages. First, despite its simplicity, it is able to learn

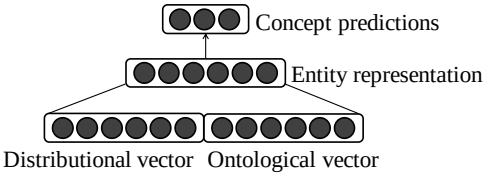

Figure 1: Graphical representation of the ONTODS model as a feedforward neural network architecture

a more complex function over the vector space than the typical candidates for inference with distributional semantics; namely, vector addition and cosine similarity. Second, we are able to train one model that jointly predicts all of the concept labels in one feedforward pass, rather than training separate classifiers for each concept. A graphical representation of the architecture of our model is presented in Figure 1.

Note that in this architecture, both the ontological features in the input vector and the predictions in the output vector refer to the membership of the entity in concept extensions. In our experiments, the features in the output vector will actually be a subset of the features in the ontological vector, because we will only use the model to make predictions about the most commonly occurring concepts. This design architecture is reminiscent of autoencoders, which have been applied to learn a compositionality function for distributional semantics (Socher et al., 2011), though in our case, the input and output vectors are not identical. Our use of regularization, weight decay, and parameter tuning on a development set prevents the model from overfitting to the training data by simply mirroring the appropriate values of the input vector.

4 Experiments

Our experiments were conducted on the collaborative knowledge base, Freebase. We extracted three classes of entities from the June 9, 2014 dump of Freebase by taking instances of top-level concepts (i.e., Freebase types) corresponding to People, Organizations, and Locations, as shown in Table 1. We chose these classes because they are the entity classes most often modelled by other work in NLP, such as by NER taggers (Finkel et al., 2005). These classes also tend to be a part of many different scenarios, thus there should be rich ontological structures to learn. In addition to the entities, we extracted all of the concepts that these entities are tagged with, in order to construct the ontological vector component of our model.

We then filtered the entities and concepts according to several frequency and quality criteria. For entities, we required the following characteristics: *(1)* there must be a word2vec vector available for that entity, as determined by a string match to the entity's name or one of its aliases; *(2)* the entity must belong to a minimum of five concepts; *(3)* the entity must satisfy a minimum frequency threshold, as follows.

We estimate the frequency of an entity by taking the frequency of the name of the entity in the Gigaword corpus. Where the name consists of multiple words, the minimum of these is taken. We used a frequency threshold of 150, which is actually quite low given the size of the Gigaword corpus. We chose to filter on frequency so that the distributional component would have seen the entity often enough to learn something useful about it.

Of the roughly one million entities in Freebase in these three categories, 84,286 entities passed the above filters.

For the concepts, we required the following characteristics: *(1)* the concept must contain a minimum of ten entity instances; *(2)* the concept must not be a /user or /m type. The second criterion removes many concepts that are overly specific and only of interest to a particular user, containing for example lists of landmarks that a user would like to visit. In addition, we removed the concept used to construct an entity category, and the concept /common/topic, because all of the entities in an entity class would be instances of these concepts. 1,262 concepts of the original 5,024 were retained after filtering.

Following filtering, the remaining entities are randomly assigned to training, development, and test sets in a 60%-20%-20% split. Table 1 provides a summary of several statistics about the data sets that we extracted.

4.1 Method

We applied the models described above to predict the concept memberships of entities in the fifty most common concepts of each entity class. We focused on the most common concepts, because they are likely to be the important high-level divisions in the entity class, and are also more likely

Entity category	**Freebase ID**	***N* entities** (train + dev + test)	***N* concepts**
People	/people/person	23053 + 7684 + 7685	530
Organizations	/organization/organization	4771 + 1591 + 1591	260
Locations	/location/location	22746 + 7582 + 7583	472

Table 1: Basic statistics concerning the subsets of Freebase that we extracted for our experiments. **Freebase ID** refers to the top-level concept used to define the entity classes that we extract. ***N*** represents the count of unique entities or concepts.

to be correctly annotated. These fifty concepts to be predicted are themselves part of the ontological vector used in the ONTO and ONTODS models. To ensure that the models do not have access to the label to be predicted at prediction time, we predict the membership for each concept separately, and mask the corresponding element of the ontological vector by setting it to zero. So, if we are predicting whether an entity is *Male*, we set the dimension corresponding to the *Male* concept in the input ontological vector to 0. We repeat this process for each concept to be predicted in the output vector. In Section 4.3, we will also test the effect of having only partial or no ontological information in the ontological vector for the ONTO and ONTODS models.

We train the feedforward neural network model by backpropagation using stochastic gradient descent with a decreasing learning rate schedule, and weight decay to prevent overfitting. To tune the parameters involved, as well as other parameters such as the number of units in the hidden layer, the amount of randomness in the initialization of the weight matrices, and the number of training epochs to perform, we use the Spearmint Bayesian optimization package (Snoek et al., 2012). We tune the parameters on the held-out development set for each entity class separately. For almost all of the models, training for 20 iterations with 100 hidden units achieves the best performance on the development set [2].

As our evaluation measure, we adopt mean average precision (MAP) from work in relation extraction and information retrieval. For each concept, the predictions from the model results in a ranking of entities that belong to the concept, in decreasing order of probability. This ranking is compared against the gold-standard extracted from FreeBase using the average precision measure:

$$AP = \frac{\sum_{k=1}^{N}(P(k) \times rel(k))}{N}, \qquad (2)$$

where $P(k)$ is the precision of the top k entities ranked by our model, $rel(k)$ is an indicator function that is 1 exactly when the kth entity is correctly predicted to be an instance of the concept, and N is the total number of entities that this concept contains. The mean average precision (MAP) is then the mean of the average precisions over all concepts. MAP is the appropriate measure for this task, as classification accuracy would give a misleading picture of performance; most entities do not belong to most concepts, so simply predicting that all entities belong to no concepts would give a very high accuracy score.

[2]The best parameter settings are available on the author's website or upon request.

	People	**Organ.**	**Loc.**
DS	45.06	43.66	38.15
ONTO	41.12	47.55	73.26
ONTODS	**50.04**	**56.60**	**75.63**

Table 2: Entity ranking results by input feature set in terms of the mean average precision measure (%). All differences are statistically significant by a randomized bootstrap test at $p < 0.0001$.

4.2 Results

The results of our concept prediction models are presented in Table 2. All differences in MAP between models trained on the same data set are statistically significant, by the randomized bootstrap method. The results show that our ONTODS model achieves the best performance on all three entity classes in terms of MAP, outperforming both the ONTO and the DS models. Comparing ONTO and DS, DS achieves better performance on People, but not on Organization, and is substantially worse on Locations.

People	Organizations	Locations
/people/deceased_person	/dining/restaurant	/architecture/venue
Benjamin Franklin 1	Cold Stone Creamery 1	Staples Center 1
Christopher Columbus 1	Rainforest Cafe 1	Candlestick Park 1
Ronald Reagan 1	Frontera Grill 1	MTS Centre 1
Duke Ellington 1	Waffle House 1	Xcel Energy Center 1
/film/music_contributor	/organization/organization_member	/geography/river
Frank Sinatra 0	MIT 1	Yamuna 1
Sean Combs 0	University of Virginia 1	Sugar Creek 1
Fred Astaire 0	University of Connecticut 0	Sugar Creek 1
Ice Cube 1	DirecTV Group 0	Brazos River 1

Figure 2: The highest-ranked entities for six select concepts according to the ONTODS model. Next to the name of the entity, a 1 indicates that the entity belongs to the concept according to Freebase, and 0 means it does not. For the river concept, Sugar Creek appears twice due to a duplicate entry in Freebase.

model: condition	**People**	**Organ.**	**Loc.**
ONTO: half	29.62	32.76	58.28
ONTO: all-but-one	41.12	47.55	73.26
ONTODS: none	32.62	40.42	27.08
ONTODS: half	44.85	48.78	65.08
ONTODS: all-but-one	**50.04**	**56.60**	**75.63**

Table 3: Entity ranking results in the partial ontological information experiment, by MAP (%). The results from "all-but-one" rows are identical to corresponding rows in Table 2.

Figure 2 shows several rankings made by the best performing ONTODS model for different concepts. Overall, almost all of the top rankings are correct according to the information extracted from Freebase. Several apparently incorrect rankings seem to be related to problems with the coverage of Freebase. For example, Frank Sinatra, Sean Combs, and Fred Astaire are not labelled as film music contributors in the version of Freebase we used. Other errors are in categories that seem to be less well-defined, such as /organization/organization_member, a concept that describes entities that belong to some other unspecified organization.

4.3 Partial Ontological Information

Earlier, we motivated the need for ontological information to model rare occurring or invented entities, yet knowledge bases are incomplete, and reliable ontological information about an entity is not always available. In this section, we simulate having partial ontological information of an input entity by masking some of the features in the ontological vector. In future work, we would like to design a system that can extract ontological information about an entity from a short passage.

Using the same trained models from the previous section, we conducted experiments in which we mask some of the input features in the ontological vector under the following three conditions, representing a decreasing amount of available information about an entity:

All-but-one This condition represents the same setting as the previous experiments, in which the model predicts the output features one at a time, and has access to all of the ontological features except for the one being predicted.

Half We ranked the output concepts by the number of entities that they contain, and then assigned them into two groups in an alternating manner. The two groups of concepts are thus roughly matched in terms of the number of entities they contain. We predict each group separately, masking those concepts in the input ontological vector; i.e., when predicting the first group of concepts, the model only has access to information about the second group of concepts, and vice versa.

None We masked all of the concepts to be predicted in the ontological vector, setting all of those features to zero. Note that the model still has access to the remaining ontological features that are not in the output vector. Thus, this setting still has access to some ontological information, unlike the DS model.

	Avg. Max. Jaccard
People	0.3525
Organizations	0.4509
Locations	0.5717

Table 4: Average maximum Jaccard similarity for the top 50 concepts in each entity class

We applied the ONTODS model under all of these conditions, and the ONTO model under the all-but-one and half conditions only, as we found that it would have very little information to make predictions on under the none condition. We used the same best performing models from the previous experiment, as the training was not affected. The results of this experiment are presented in Table 3. Unsurprisingly, the performance of both models degrade substantially when given only partial ontological information. Note, however, that the ONTODS model in the half condition is still better than the DS and ONTO models in the all-but-one condition on two of the three entity classes.

4.4 Discussions

What accounts for the differing contributions of the ontological and the distributional components to the performance for the different entity classes? In particular, ontological information seems to be especially important for the Locations entity class, whereas distributional information seems to be better for the People entity class. We consider the correlations between the different concepts as an explanation for this result. Intuitively, the greater the correlations between the concepts for a certain entity class, the more useful ontological information is in making inferences about concept memberships of entities.

We compute a measure based on Jaccard similarity between the concepts for this analysis. For each of the top 50 concepts represented in the output vector, we find the maximum Jaccard similarity between that concept and the other concepts in the training set:

$$\textit{maxJ}(c) = \max_{c'} \frac{|\textit{exten}(c) \cap \textit{exten}(c')|}{|\textit{exten}(c) \cup \textit{exten}(c')|}. \quad (3)$$

Then, we take the average of this maximum Jaccard similarity over the top 50 concepts. We use the maximum similarity to other concepts rather than the average; the average similarity could be low due to having many unrelated concepts, which a statistical learner would identify as irrelevant. Across the three entity classes, the ranking of the average maximum Jaccard similarity matches the apparent importance of the ontological component of the models in the entity ranking task (Table 4). This result provides an explanation for the different performances of the models in the entity ranking task, and could be used to approximate model performance given a new data set.

5 Deriving Semantic Relations

We further analyze our model's performance by examining its ability to recognize semantic relations between concepts. This analysis is not a formal evaluation of the models, but serves two purposes. First, it is a qualitative test of the entity rankings of our model. Second, it demonstrates inferences that follow directly from concept extension predictions without the need to train yet another special-purpose classifier, for example to determine hypernymy or synonymy.

Whereas relations such as hypernymy and synonymy follow directly from crisp, 0-1 concept extensions predictions, we choose instead to use the ranking probabilities that are the output of our model. This avoids issues with choosing an appropriate cut-off for the predictions, and also allows the models to make soft predictions of lexical semantic relations between concepts. We focus below on hypernym/hyponym relations; because Freebase explicitly attempts to standardize and canonicalize all entities and types, we do not expect to find good synonyms.

We thus view the predictions produced by the models as fuzzy sets (Zadeh, 1965)[3], and use standard operations from fuzzy set logic to determine hypernymy. Our models above learn a function $\vec{y} = f(\vec{x})$, where y_k is the probability that the input entity belongs to concept c_k. For a given concept c_k, let us now aggregate the model predictions over all entities into a vector $\mu_k(x)$, which has a length equal to the number of entities in the data set. This can be seen as a membership function of a fuzzy set that provides a score between 0 and 1 of an entity x in $\textit{exten}(c_k)$. We use the follow-

[3]We leave aside the philosophical issue of whether our models' output values should be interpreted as probabilities of set membership or degrees of set membership.

c_i	c_j	$\subseteq$	$\supseteq$	c_i	c_j	$\subseteq$	$\supseteq$
People ONTODS				**People** DS			
tv_program_guest	/film/actor	0.99	0.35	cricket_bowler	cricket_player	0.99	0.68
theater_actor	/film/actor	0.99	0.41	olympic_athlete	pro_athlete	0.98	0.18
celebrity	/film/actor	0.95	0.43	football_player	pro_athlete	0.97	0.18
Organizations ONTODS				**Organizations** DS			
venture_company	employer	1.0	0.08	airline	employer	0.99	0.03
football_team	sports_team	0.99	0.22	airline	aircraft_owner	0.98	0.91
restaurant	employer	0.99	0.05	university	educ_inst	0.92	0.85
Locations ONTODS				**Locations** DS			
river	geog_feature	0.99	0.28	capital_admin_div	stat_region	0.99	0.09
river	body_of_water	0.97	0.38	university	educ_inst	0.95	0.91
body_of_water	geog_feature	0.97	0.70	building	structure	0.96	0.57

Figure 3: Scores for several subset and superset relations learned by two of our models using fuzzy set logic operations. The $\subseteq$ columns display the score $\mathit{hypo}(c_i, c_j)$, while the $\supseteq$ columns display $\mathit{hypo}(c_j, c_i)$. We have abbreviated several concept names for space reasons.

ing definitions of intersection and union between fuzzy sets A and B:

$$\mu_{A\cap B} = \min(\mu_A, \mu_B) \quad (4)$$

$$\mu_{A\cup B} = \max(\mu_A, \mu_B). \quad (5)$$

A concept c is a hyponymy of another concept c' if $\mathit{exten}(c) \subseteq \mathit{exten}(c')$. We determine the subset relation in fuzzy logic reducing it to fuzzy set intersection and set equality, and we determine fuzzy set equality by using a generalized version of Jaccard similarity using L1-norms:

$$A \subseteq B \leftrightarrow A \cap B = A \quad (6)$$

$$\mathit{fuzzyJ}(\mu_A, \mu_B) = \frac{\|\mu_{A\cap B}\|_1}{\|\mu_{A\cup B}\|_1}. \quad (7)$$

The degree of hyponymy of c_i to c_j, $\mathit{hypo}(c_i, c_j)$, is then simply $\mathit{hypo}(c_i, c_j) = \mathit{fuzzyJ}(\mu_{i\cap j}, \mu_i)$.

We present several subset relations discovered by the ONTODS and DS models in Figure 3, as indicated by a high *hypo* score between the concepts. We chose these models because the former is the best-performing model in entity ranking, and the latter does not include ontological information in its entity representation. This method finds several good hyponym/hypernym relations, such as football_team $\subseteq$ sports_team, and restaurant $\subseteq$ employer. It also finds chains of relations, such as sports_facility $\subseteq$ venue $\subseteq$ structure, and river $\subseteq$ body_of_water $\subseteq$ geographical_feature.

6 Conclusion

We have argued that concept extensions can form the basis of a vector-space model of semantics. Our model learns entity representations by combining ontological information derived from a knowledge base with distributional information trained to predict concept extensions. Our experiments indicate the success of this model, and we perform several analyses to explain the relative importance of the ontological and distributional semantic components of our model, as well as the ability of the model to recover semantic relations between concepts using fuzzy set logic.

Learning concept extensions provides a method to integrate distributional semantics with formal, compositional semantics. For example, semantic relations between concepts could be detected based on their formal, set-theoretic definitions, as shown in Section 5. The framework and model presented in this paper suggest a natural way to predict these and other semantic relations without the need for another classification step.

It would also be interesting to see whether the ontological information/concept extensions, which in this work was supplied by a knowledge base, could be derived or augmented through other means, such as by using image data (Young et al., 2014).

Acknowledgments

We would like to thank Patricia Araujo Thaine, Aida Nematzadeh, Nissan Pow, and the anonymous reviewers for useful discussions and feedback. This work is funded by the Natural Sciences and Engineering Research Council of Canada.

References

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 385–393. Association for Computational Linguistics.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June. Association for Computational Linguistics.

Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. 2013. Montague meets Markov: Deep semantics with probabilistic logical form. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM-2013)*.

Kurt 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 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

Razvan C. Bunescu and Raymond J. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, volume 45, pages 576–583.

Jenny R. 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, June. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Daniel Fried and Kevin Duh. 2014. Incorporating both distributional and relational semantics in word representations. *arXiv preprint arXiv:1412.4369*.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Mapping conceptual features to referential properties. In *Proceedings of the 3rd International ESSENCE Workshop: Algorithms for Processing Meaning*.

Aurélie Herbelot and Mohan Ganesalingam. 2013. Measuring semantic content in distributional vectors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 440–445, Sofia, Bulgaria, August. Association for Computational Linguistics.

Paul McNamee and Hoa T. Dang. 2009. Overview of the TAC 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, volume 17, pages 111–113.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. 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, June. Association for Computational Linguistics.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959.

Richard Socher, Eric H. Huang, Jeffrey Pennin, Christopher D. Manning, and Andrew Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809.

Ichiro Yamada, Kentaro Torisawa, Jun'ichi Kazama, Kow Kuroda, Masaki Murata, Stijn De Saeger, Francis Bond, and Asuka Sumida. 2009. Hypernym discovery based on distributional similarity and hierarchical structures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 929–937, Singapore, August. Association for Computational Linguistics.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Lotfi A. Zadeh. 1965. Fuzzy sets. *Information and Control*, 8(3):338–353.