Formal-Logical Distributional Semantics: Applications to Property Inference

Michael Sullivan

Department of Linguistics, University at Buffalo
mjs227@buffalo.edu

Abstract

In this extended abstract, I propose the Formal-Logical Distributional Semantics (FoLDS) model, which generates complex-valued word vectors drawn from a fuzzy-logical model world imperatively constructed from logical-form representations of sentences from Simple English Wikipedia articles. A complex-valued similarity metric arises naturally from complex-valued embeddings, which permits FoLDS to leverage multiple axes of similarity simultaneously (antonymy/synonymy and relatedness). Moreover, I argue that using a rule-based parser to translate sentences into logical forms has a syntactic de-noising effect on the resulting embeddings, which allows FoLDS to effectively leverage a smaller training corpus. I show that FoLDS is able to achieve near-state-of-the-art results (within 10%) on a property inference task despite using word embeddings obtained from a corpus approximately two hundred times smaller than the training corpora that competing approaches use.

Introduction

Consider the following passage: "Many well-read adults know that Buddha sat long under a banyan tree [...] and Tahitian natives lived idyllically on breadfruit and poi" (Levy and Nelson 1994). Even if one has never heard the terms *banyan tree*, *breadfruit*, and *poi*, observing them in this single linguistic context suffices to infer some of their properties; a banyan tree must be somewhat large (as Buddha was able to sit under one), and breadfruit and poi must be foods. Note that fluent speakers are able to make such inferences without having any knowledge grounding these terms to real-world concepts.

This phenomenon, in which language users are able to infer properties of words purely from their linguistic distributions, is known as *property inference* (Li and Summers-Stay 2019). Property inference is a logic-oriented task (Patalano, Wengrovitz, and Sharpes 2009); given that an aardwolf is a type of animal, we assume that *aardwolf* has all of the properties that *animal* has (*alive*, *breathes*, etc.)—here, we are reasoning from *hyponymy* (Herbelot and Vecchi 2015). Given that *alligator* has the property *is-dangerous*, we assume that *rabbit* does not have that property—here, I argue, we are reasoning from *antonymy*.

Previous approaches to this task—such as that of Rosenfeld and Erk (2022), which achieved the current state-of-the-art (SoTA) property inference results—have focused on inferring properties from distributional word embeddings. In this paper, I argue that traditional distributional encodings—real-valued vectors obtained from surface text—are insufficient to fully model the kind of inductive reasoning necessary for property inference. I propose that appropriate distributional encodings for property inference must permit a two-dimensional similarity metric capable of expressing two axes of comparison: *synonymy/antonymy* and *relatedness* (similar to hyponymy). Furthermore, even advanced language models such as BERT (Devlin et al. 2018) have been shown to be easily confused by syntactic paraphrases such as passivization (Chaves and Richter 2021). To mitigate this issue, I draw word embeddings from logical forms; translating sentences in a corpus into logical representations has the effect of equivalence-classing syntactic paraphrases.

In this work, I propose the Formal-Logical Distributional Semantics (FoLDS) model, which involves obtaining complex-valued word vectors drawn from a fuzzy-logical model world imperatively constructed from logical-form representations of sentences, which I generate by applying the English Resource Grammar (ERG; Copestake and Flickinger 2000) parser to Simple English Wikipedia[2] (SEW). Such representations are inherently sensitive to negation (and therefore antonymy) and other logical operators, and insensitive to syntactic periphrases (as they are drawn from logical forms). Complex-valued vectors naturally give rise to a complex-valued similarity metric, which is able to leverage these representations' sensitivity to negation to express a measure of *synonymy/antonymy* as well as *relatedness* within a single complex number. I show that FoLDS is able to achieve near-SoTA results (within 10%) despite using distributional count vectors obtained from a corpus approximately two hundred times smaller than the training data that Rosenfeld and Erk use in their analysis.

Accepted to Workshop on Knowledge Augmented Methods for Natural Language Processing, in conjunction with AAAI 2023.

[2]https://simple.wikipedia.org

Feature	Value
a-utensil	0.634 (19/30)
found-in-kitchens	0.600 (18/30)
used-with-forks	0.534 (16/30)
a-cutlery	0.500 (15/30)
is-dangerous	0.467 (14/30)
a-weapon	0.367 (11/30)

Table 1: McRae et al. (2005) feature norms for the concept *knife*. For all other features Q, $F(knife)_Q = 0$.

Previous Work

Rosenfeld and Erk (2022) provide a comprehensive discussion of other work on property inference tasks, and I refer readers to their paper for a more in-depth discussion. For the sake of brevity, I discuss only the most relevant property inference methods that they analyze. I compare the results that FoLDS obtains to those of Rosenfeld and Erk on the McRae et al. (2005) feature norm database; I provide a more in-depth discussion of that aspect of the authors' work.

Task Design

A *feature norm database* consists of a set of *concepts* (words) and a set of *features*, in which each concept w is assigned a feature vector $F(w) \in \mathbf{R}^n$, where n is the number of features in the database. The value of $F(w)_Q$ is the value of the feature Q for the word w. For example, the McRae et al. (2005) database, which I use to evaluate FoLDS, consists of 541 concepts and 2526 features; feature values are obtained from experiment participants' judgements.

Rosenfeld and Erk (2022) create ten random folds consisting of 50 concepts each from the dataset. On each fold, the concepts within the fold represent the set U of *unknown* words—words which have been observed in text but are not grounded to real-world concepts—and the concepts outside of the fold represent the set K of *known* words. For each unknown word $u \in U$, the feature vector $F(u)$ is zeroed out; the task is to reconstruct $F(u)$ given the known features in K and the similarity between u and each word in K.

Previous Property Inference Methods

Rosenfeld and Erk (2022) examine a wide variety of property inference methods in their analysis. Note that all of the property inference methods evaluated by those authors share the same distributional word embeddings; LSA vectors drawn (context window of two) from four different corpora ($\sim$4.2 billion words total): ukWaC (Ferraresi et al. 2008), Google Gigaword (Graff and Cieri 2003), Wikipedia[3], and BNC (BNC 2007). What varies across methods is how they use these embeddings to estimate properties.

A variant of the Modified Adsorption (ModAds) algorithm (Talukdar and Crammer 2009) achieves SoTA results in Rosenfeld and Erk's analysis. I refer the interested reader to Talukdar and Crammer's and Rosenfeld and Erk's papers for an in-depth explanation of ModAds and its application to property inference tasks.

[3]https://en.wikipedia.org

Many of the property inference methods that Rosenfeld and Erk examine include a *shifted* variant. This does not indicate a difference in the models' architectures, but rather the feature vectors $F(w)$. In the *shifted* trials, Rosenfeld and Erk decrease the values of those properties Q such that $F(w)_Q = 0$ to *negative* values; this is to increase the separation between irrelevant and relevant properties.

FoLDS

To motivate FoLDS, I first discuss the relevant drawbacks of traditional distributional embeddings with respect to property inference tasks. The remainder of this section is a mostly conceptual overview of the FoLDS architecture—for the sake of brevity I refer readers to the appendix for a mathematical description of the algorithms involved.[4]

Motivation

First, to fully capture meaning, at least sufficient aspects of meaning for the purposes of property inference, I argue that co-occurrence statistics should be equivariant with respect to syntactic paraphrases (non-canonical constructions such as passivization, topicalization, etc.; Colin and Gardent 2018). For example, the active sentence in example (1a) should belong to the same equivalence class as its passive counterpart in (1b).

(1a) *Tahitian natives feasted on* ___
(1b) ___ *was feasted on by Tahitian natives*

Examples (1a-b) are *not* the same context when a distributional model only has access to surface forms (i.e. raw text). Converting both (1a-b) to a first-order logic (FOL)-typed λ-calculus representation, on the other hand, yields the exact same formula: $\lambda x.feast\text{-}on(tahitian\text{-}natives, x)$.

Intuitively, treating logical formulae as distributional contexts should decrease the amount of training data required to obtain accurate embeddings, as the larger amounts of data required to learn to equate passive constructions and their active counterparts are no longer necessary. Additionally, Herbelot and Copestake (2021) demonstrate that distributional embeddings obtained from logical-form descriptions of model-theoretic representations of events and situations are highly effective at modeling elementary semantic relations such as hyponymy, synonymy, and antonymy.

Second, the two sentences in examples (2a-b) *increase* the (traditional) distributional similarity between *herbivore* and *carnivore* (many of the words in those two sentences are the same), when in fact they are antonymous.

(2a) *Alligators are not ever considered herbivores, even when food is scarce*
(2b) *Alligators are always considered carnivores, even when food is scarce*

However, while (2a-b) suggest that *herbivore* and *carnivore* are antonymous, it does not indicate that they are *unrelated*. Inferences can still be drawn from antonymous but

[4]Code available on GitHub: https://github.com/mjs227/FoLDS

related categories; given that *carnivore* has a property such as *eats-meat*, a property inference method should be able to leverage the antonymy between *carnivore* and *herbivore* to hypothesize that *herbivore* does not have that property. Traditional distributional models such as BERT are insensitive to negation and word order (Ettinger 2020), and so inherently less capable of detecting the evidence of antonymy between *herbivore* and *carnivore* that (2a-b) contribute.

Third, there is a distinction to be drawn between synonymy/antonymy and relatedness. Unrelated categories should not contribute to property inference; for example, properties of *duct tape* should have no bearing on a property inference method's estimation of the properties of *alligator*.

The latter two points suggest that property inference requires a two-dimensional similarity metric that measures both synonymy/antonymy and relatedness.

Architecture

To convert SEW into logical representations, I first pass each article through Spacy's *NeuralCoref* coreference resolution module[5] and *SentenceRecognizer* sentence-segmentation pipeline[6]. I then apply the *ACE ERG* parser[7] to each sentence to obtain its *Minimal Recursion Semantics* (MRS; Copestake et al. 2005) representation. I use the coreference data to equivalence-class co-referring entities and quantifiers in the MRS representations. I then use a heuristic procedure to resolve quantifier scope and convert the MRS structures into a representation similar to FOL.

I construct a fuzzy-logical model world using the FOL-like representations of each sentence in SEW. Non-quantified formulae immediately assign properties to entities, and existential quantifiers are removed by inserting *dummy entities*—for example, $\exists x[blue(x) \wedge car(x)]$ becomes $blue(d_n) \wedge car(d_n)$. I remove universal quantifiers by assigning the properties in the scope of a universally-quantified formula to all entities that satisfy its restriction. For example, $\forall x[car(x) \rightarrow blue(x)]$ becomes $blue(x_1) \wedge \cdots \wedge blue(x_n)$, where $\{x_1, \ldots, x_n\}$ is the set of all entities that satisfy the formula $\lambda x.car(x)$. The procedure applies recursively to complex formulae until all logical operators (except negation) are removed from the structure. This process yields a set of 4-tuples (ϕ, x, p, n), where ϕ is a λ-abstracted formula, x is an argument, p is the *positive* occurrence count of $\phi(x)$, and n is the *negative* count.

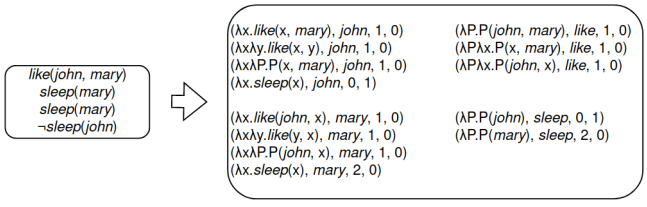

Figure 1: An example of converting logical formulae to distributional contexts.

[5]https://spacy.io/universe/project/neuralcoref

[6]https://spacy.io/api/sentencerecognizer

[7]http://sweaglesw.org/linguistics/ace/

For example, $(\lambda x.like(x, mary), john, 1, 0)$ encodes the fact that there is one positive occurrence (and zero negative occurrences) of $john$ liking $mary$. On the other hand $(\lambda x.sleep(x), john, 0, 1)$ encodes the fact that there is one *negative* occurrence (and zero positive occurrences) of $john$ sleeping—i.e. one count of him *not* sleeping.

These 4-tuples are then used to construct complex-valued count vectors; contexts (λ-abstracted formulae) index the dimensions, whose values are complex numbers of the form $a+bi$, where a is the positive count and b the negative count. FoLDS calculates the similarity between any two such entity vectors using the following equation:

$$sim(x, y) = \tau(x, y) \cdot \Omega(x, y) \quad (1)$$

The term $\tau(x, y)$ is a length-one normalized complex number—this is a measure of the *synonymy* between x and y, and will fall in the *positive* quadrant of the complex plane. Given two entities x, y that are completely synonymous, the value of $\tau(x, y)$ will be $1 + 0i$. Here *synonymous* means that for any context ϕ, the truth value of x_ϕ equals the truth value of y_ϕ (*truth value* does not refer to the actual complex number occupying that coordinate, but rather its distance [roughly] from the real axis). On the other hand, x and y are considered *antonymous* ($\tau(x, y) = 0+1i$) if they *disagree* on every context that they have in common. For a given context ϕ, let $r(x_\phi)$ denote the truth value of x_ϕ. Then x and y are considered antonymous if, for each context ϕ, $r(x_\phi) = 1 - r(y_\phi)$. Values lying between these two extrema reflect graded degrees of synonymy/antonymy.

Finally, the length-one complex number $\tau(x, y)$ is scalar multiplied by the real number $\Omega(x, y) \in [0, 1]$ to obtain $sim(x, y)$. $\Omega(x, y)$ is an asymmetric measure of the overlap between the contexts in which x and y appear. For example, $\Omega(apple, fruit)$ should be close to 1, but $\Omega(fruit, apple)$ should be closer to 0—every property that *fruit* has is also a property of *apple*, but not vice versa.

However, the procedure described above only yields vectors for each *entity*, not each *word*. For each of the concept words w in the McRae et al. (2005) database, I obtain an embedding by summing together the entity vectors for each entity in $\{e \mid \exists z \in \mathbf{R}_+[e_w = z + 0i]\}$—the set of all entities that have a nonzero *positive* count, and a zero *negative* count for the context $\lambda x.w(x)$.

Experiment and Results

The real-valued feature vectors $F(w)$ must be converted to complex-valued vectors $C(F(w))$ in order to interface with the similarity metric (Equation 1) in a meaningful way. Given a (real-valued) feature vector $F(w)$ for some word w: for each property Q, if $F(w)_Q = 0$, then $C(F(w))_Q = 0 + 1i$. Otherwise, $C(F(w))_Q$ lies on the positive quadrant of the unit circle, its angle placed between $0°$ and $45°$, inversely proportional to the value of $F(w)_Q$. This is intended to mimic the *shifting* procedure discussed above.

To estimate properties, I mimic the Johns and Jones (JJ; 2012) method (see appendix for details), but with complex rather than real numbers. I found that only considering the set $K_n(u)$ of top n most *related* (determined by the magnitude $\Omega(u, w)$) known words w to a given unknown word u

yields the best results. Via grid search, I found $n = 25$ to be the optimal value for this experiment.

For a given unknown word u and property Q, the estimated (complex) value of Q for u, $P(u)_Q$, is calculated as follows:

$$P(u)_Q = \sum_{w \in K_n(u)} abs(C(F(w))_Q \cdot sim(u, w)) \quad (2)$$

Where $abs(a + bi) = |a| + |b|i$. This function forces the resulting value of $C(F(w))_Q \cdot sim(u, w)$ to the positive quadrant of the complex plane while preserving its distance from the real axis.

Recall that $sim(u, w)$ is the product of the length-one complex number $\tau(u, w)$ and the real scalar $\Omega(u, w)$, which reflect synonymy/antonymy and relatedness, respectively. If u and w are synonymous, then $\tau(u, w) = 1+0i$. Given some property of w (represented by a positive-quadrant complex number $a + bi$) whose value is unknown for u, $(a + bi) \cdot \tau(u, w) = a + bi$: synonymous words are predicted to have the same values for each property. On the other hand, suppose that u and w are antonymous, so that $\tau(u, w) = 0 + 1i$, which corresponds to a 90° rotation: antonymous words are predicted to have opposite property values. The real number $\Omega(u, w)$ scales $\tau(u, w)$: those known words w with higher values of $\Omega(u, w)$ (i.e. which are more related to u) will contribute more to the overall inference than those with lower values (see Figure 2).

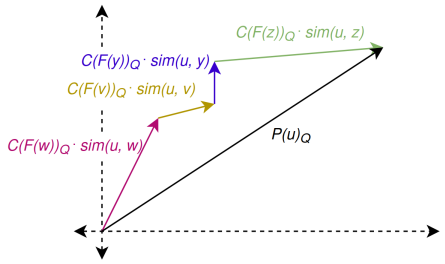

Figure 2: Equation 2 can be viewed as the average of the predicted values of Q for u for each known word w, weighted by their overlap ($\Omega(u, w)$).

In order to compare the predicted values $P(u)$ to the ground-truth values $F(u)$, each $P(u)_Q$ must be converted to a real number $r(P(u))_Q$. I leave the details of this process to the appendix; for the purposes of this discussion it suffices to state that as $P(u)_Q$ approaches the real axis, $r(P(u))_Q \to 1$, and as $P(u)_Q$ approaches the imaginary axis, $r(P(u))_Q \to 0$.

Rosenfeld and Erk (2022) use two separate evaluation metrics in their analysis. The first, *Mean Average Precision* (MAP; Zhu 2004) measures a given property inference method's ability to rank *relevant* features above *irrelevant* features. The order (ranking) in which these predicted features are returned does not impact the MAP score. Additionally, the predicted property values for ground-truth irrelevant features (i.e. those which have a value of zero in the feature norm database) do not impact the MAP score, as long as the ground-truth relevant features are ranked higher than the irrelevant ones. I do not consider MAP to be an effective evaluation metric for property inference tasks; it essentially evaluates a different task in which property ranking has been reduced to a 0/1 distinction. I do not discuss MAP further.

Method	ρ	**Method**	ρ
Property frequency	0.049	Property sum	0.042
JJ (1 step)	0.114	JJ (2 step)	0.107
Linear SVM	0.077	Linear SVM shifted	0.089
Cosine SVM	0.082	Cosine SVM shifted	0.082
Linear PLS	0.077	Linear PLS shifted	0.075
Cosine PLS	0.082	Cosine PLS shifted	0.083
ModAds equal	0.161	ModAds equal shifted	0.244
ModAds decay	0.161	ModAds decay shifted	0.243
ModAds NN	0.244	ModAds NN shifted	**0.281**
FoLDS	0.253		

Table 2: Comparison of methods in Rosenfeld and Erk (2022) against FoLDS on the McRae et al. database.

The second evaluation metric that Rosenfeld and Erk use in their analysis, Spearman's ρ, measures the correlation between the *rankings* of ground-truth and predicted properties (Dodge 2008). Many deficiencies of the MAP score do not pertain to this metric; any discrepancy between the relative ranking of predicted properties and that of the ground-truth properties will negatively impact the ρ score. Following Rosenfeld and Erk, I average over all Spearman ρ scores for each unknown word in each fold for evaluation. FoLDS achieves a Spearman ρ of **0.253**, which is the second-best out of the 18 methods in Rosenfeld and Erk's analysis (see Table 2). Crucially, all of these methods use LSA vectors generated from a PPMI-transformed co-occurence matrix (Roller, Erk, and Boleda 2014) obtained from a lemmatized and POS-tagged 4.2 billion word corpus, while FoLDS uses count vectors obtained from a 24.5 million word corpus (∼200 times smaller).

Conclusion

In this paper, I proposed the use of FoLDS embeddings for property inference tasks, and demonstrated that this method achieves near-SoTA Spearman ρ results using significantly less training data than competing approaches. I argued that translating sentences into logical forms has a syntactic denoising effect on the resulting embeddings, allowing FoLDS to effectively leverage its smaller training corpus. Moreover, a complex-valued similarity metric arises naturally from complex-valued embeddings, permitting FoLDS to leverage two axes of similarity. In future work, I will utilize a larger training dataset with the goal of further improving performance on the McRae et al. (2005) database and other property inference tasks. Additionally, I am in the process of replicating the methods that Rosenfeld and Erk (2022) use in their analysis with GloVe (Pennington, Socher, and Manning 2014) and LexVec (Salle, Villavicencio, and Idiart 2016) embeddings in order to establish a more recent-embedding based baseline. I will also evaluate FoLDS on other NLP tasks, including *question-answering* (e.g. Rajpurkar et al. 2016) and *semantic textual similarity* (e.g. Cer et al. 2017).

References

BNC. 2007. The British National Corpus, Version 3. Distributed by Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium.

Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *Eleventh International Workshop on Semantic Evaluations*.

Chaves, R. P.; and Richter, S. N. 2021. Look at that! BERT can be easily distracted from paying attention to morphosyntax. *Proceedings of the Society for Computation in Linguistics*, 4(1): 28–38.

Colin, E.; and Gardent, C. 2018. Generating syntactic paraphrases. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 937–943.

Copestake, A.; and Flickinger, D. 2000. An Open Source Grammar Development Environment and Broad-coverage English Grammar Using HPSG. *LREC*, 591–600.

Copestake, A.; Flickinger, D.; Pollard, C.; and Sag, I. A. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2): 281–332.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dodge, Y. 2008. *The concise encyclopedia of statistics*. Springer Science & Business Media.

Ettinger, A. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8: 34–48.

Ferraresi, A.; Zanchetta, E.; Baroni, M.; and Bernardini, S. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, 47–54.

Graff, D.; and Cieri, C. 2003. English gigaword corpus. *Linguistic Data Consortium*.

Herbelot, A.; and Copestake, A. 2021. Ideal Words. *KI-Künstliche Intelligenz*, 35(3): 271–290.

Herbelot, A.; and Vecchi, E. M. 2015. Building a shared world: Mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 22–32.

Johns, B. T.; and Jones, M. N. 2012. Perceptual inference through global lexical similarity. *Topics in Cognitive Science*, 4(1): 103–120.

Levy, E.; and Nelson, K. 1994. Words in discourse: A dialectical approach to the acquisition of meaning and use. *Journal of child language*, 21(2): 367–389.

Li, D.; and Summers-Stay, D. 2019. Mapping distributional semantics to property norms with deep neural networks. *Big Data and Cognitive Computing*, 3(2): 30.

McRae, K.; Cree, G. S.; Seidenberg, M. S.; and McNorgan, C. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37(4): 547–559.

Patalano, A. L.; Wengrovitz, S. M.; and Sharpes, K. M. 2009. The influence of category coherence on inference about cross-classified entities. *Memory & cognition*, 37(1): 21–28.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392.

Robertson, S. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation*.

Roller, S.; Erk, K.; and Boleda, G. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, 1025–1036.

Rosenfeld, A.; and Erk, K. 2022. An analysis of property inference methods. *Natural Language Engineering*, 1–27.

Salle, A.; Villavicencio, A.; and Idiart, M. 2016. Matrix Factorization using Window Sampling and Negative Sampling for Improved Word Representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 419–424. Association for Computational Linguistics.

Talukdar, P. P.; and Crammer, K. 2009. New regularized algorithms for transductive learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 442–457. Springer.

Zhu, M. 2004. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2(30): 6.

Appendix

I organize the appendix into four subsections: the first covers the Johns and Jones (2012) method, the second the similarity metric (Equation 1), the third the complex-valued feature vectors $C(F(w))$, and the fourth the procedure for converting complex-valued feature estimates $P(u)_Q$ to real numbers $r(P(u))_Q$.

The Johns and Jones (2012) Method

This method estimates property values for unknown words simply as the sum of all of the property values of the known words, weighted by their cosine similarity with the unknown word in question. Let K denote the set of known words, u denote a given unknown word, and Q denote a given property. Then the estimated value of Q for u, $P(u)_Q$, is calculated as follows:

$$P(u)_Q = \sum_{w \in K} F(w)_Q \cdot cos(u, w)^\lambda \tag{3}$$

Where λ is a hyperparameter—higher values of λ reduce the influence of less similar words.

Calculating the Similarity Metric

$$sim(x, y) = \tau(x, y) \cdot \Omega(x, y) \tag{4}$$

$$\tau(x, y) = \frac{\tau'(x, y)}{|\tau'(x, y)|} \tag{5}$$

$$\tau'(x, y) = \sum_{\phi} (\sigma(\phi, x, y) \cdot \iota(\phi)) + (1 - \sigma(\phi, x, y))i \tag{6}$$

$$\sigma(\phi, x, y) = max(r(x_\phi), 1 - |r(x_\phi) - r(y_\phi)|) \tag{7}$$

$$r(a + bi) = \frac{a}{a + b} \tag{8}$$

$$\Omega(x, y) = \frac{\sum_\phi min(|x_\phi|, |y_\phi|) \cdot \iota(\phi)}{\sum_\phi |x_\phi| \cdot \iota(\phi)} \tag{9}$$

$$\iota(\phi) = log_2 \frac{\mu_{max}}{\mu(\phi)} \tag{10}$$

$$\mu(\phi) = \sum_x |x_\phi| \tag{11}$$

$$\mu_{max} = max_\phi(\mu(\phi)) + 1 \tag{12}$$

As stated above, $sim(x, y)$ (Equations 1, 4) is the product of the length-one normalized complex number $\tau(x, y)$ (Equation 5) scalar multiplied by the (positive) real number $\Omega(x, y)$ (Equation 9).

As shown in Equation 5, $\tau(x, y)$ is simply the length-one normalization of $\tau'(x, y)$ (Equation 6). To obtain $\tau'(x, y)$, FoLDS first sums over all contexts ϕ to calculate $\sigma(\phi, x, y)$ (Equation 7).

Turning to Equation 7, $\sigma(\phi, x, y)$ is designed to model fuzzy-logical implication. Note that $\sigma(\phi, x, y) = 1$ if $r(x_\phi) = r(y_\phi)$ or $r(x_\phi) = 1$, where $r(x_\phi)$ and $r(y_\phi)$ (Equation 8) denote the *truth values* of x_ϕ and y_ϕ, respectively. On the other hand, $\sigma(\phi, x, y) = 0$ if $r(x_\phi) = 0$ and $r(x_\phi) = 1 - r(y_\phi)$ (i.e. if $r(y_\phi) = 1$). So $\sigma(\phi, x, y)$ is a real number which represents the (fuzzy) degree to which $r(y_\phi)$ *implies* $r(x_\phi)$.

Returning to Equation 6, $\tau'(x, y)$ yields a complex number whose real value corresponds (roughly) to the degree to which $r(y_\phi)$ implies $r(x_\phi)$, and whose imaginary value corresponds (again, roughly) to the degree to which $r(y_\phi)$ *does not* imply $r(x_\phi)$, for all contexts ϕ.

Note that in each summand in Equation 6, the real part is calculated as the product of $\sigma(\phi, x, y)$ and $\iota(\phi)$ (Equation 10). The value of $\iota(\phi)$ is intended to mimic *inverse document frequency* (Robertson 2004) weighting, where the notion of document frequency has been replaced with sum of the magnitudes of each coordinate of the context vector ϕ. Essentially, the more frequently that the context ϕ appears, the lower the value of $\iota(\phi)$ will be.

The scalar (i.e. real) value $\Omega(x, y)$ (Equation 9) is obtained by summing together $min(|x_\phi|, |y_\phi|) \cdot \iota(\phi)$ for each context ϕ, then dividing the resulting value by $\sum_\phi |x_\phi| \cdot \iota(\phi)$ (where $|x_\phi|$ denotes the magnitude of the complex number x_ϕ). To view this conceptually, let $E(x) = \{\phi \mid x_\phi \neq 0 + 0i\}$—the set of all contexts ϕ whose value for x is nonzero (i.e. known). Then as $|E(x) \cap E(y)| \to 0, \Omega(x, y) \to 0$, and as $|E(x) \cap E(y)| \to |E(x)|, \Omega(x, y) \to 1$. The idea here is that the closer that $E(x)$ is to a subset of $E(y)$, the more confident we can be that properties of y apply to x, and the further $E(x)$ is from a subset of $E(y)$, the less confident we are that properties of y apply to x. The weights $\iota(\phi)$ ensure that less frequent contexts are more important to the value of $\Omega(x, y)$; if x and y both have known values for a very frequent context, that does not give very much evidence about the transferability of properties from y to x.

Generating Complex-Valued Feature Vectors

$$C(F(w))_Q = \begin{cases} 0 + 1i & \text{if } F(w)_Q = 0 \\ \beta(F(w)_Q) & \text{otherwise} \end{cases} \tag{13}$$

$$\beta(z) = \frac{(1 + z) + (1 - z)i}{|(1 + z) + (1 - z)i|} \tag{14}$$

Converting Complex Estimates to Real Values

Recall that the values $C(F(u))_Q$ are *shifted*, which I account for by placing a *floor* on the values of $r(P(u))_Q$. In this experiment, I found via grid search that *floor* = 0.15 yields the best results. The floor is implemented as follows:

$$r(P(u))_Q = \begin{cases} r(P(u)_Q) & \text{if } floor \leq r(P(u)_Q) \\ 0 & \text{otherwise} \end{cases} \tag{15}$$