Feature Ablation for Preposition Disambiguation

Ken Litkowski
CL Research
9208 Gue Road
Damascus, MD 20872 USA
ken@clres.com

Abstract

The development of classification models for preposition disambiguation involves the generation of thousands of features describing the context of the preposition. The best modeling technique, support-vector machines, produce weights for each feature, but these weights are difficult to interpret and use for determining the most important features. One technique that can aid in this identification is feature ablation, removing features and assessing the effect on classification performance. We build upon standard approaches for feature ablation, removing feature sets one at a time, performing upwards of 5000 iterations for each preposition. We describe our algorithm in detail, including the detailed results that are generated with each iterate. We examine these results, suggesting that intuitions about how to describe preposition behavior might not hold. In particular, factors other than the syntactic and semantic properties of the complement and the governor frequently emerge as important.

In this paper, we describe an algorithm for feature ablation using support-vector machine (SVM) modeling. In section 1, we provide background for SVM modeling used in preposition disambiguation, particularly identifying the types of features that are used. In section 2, we describe the algorithm that drills down through word-finding features, syntactic and semantic characterization features, and combinations of the two types. We describe the criteria used to identify feature sets to be ablated and the measures that are generated. In section 3, we examine these results and measures and how they might be used in characterizing preposition behavior. In section 4, we interpret the results and discuss the need for further investigations to use the results to aid in describing preposition behavior.

1. Features Used in SVM Modeling for Preposition Disambiguation

As described in Litkowski (2014) and Litkowski (2016), the Pattern Dictionary of English Prepositions (PDEP) has processed 81509 sentences in three corpora using a lemmatizer, part-of-speech tagger, and dependency parser (Tratz and Hovy, 2011). Using the parse results, in CoNLL-X format, features are extracted to describe the context of a specified preposition in each sentence. Each feature consists of three components, a word position relative to the prepositions, a syntactic or semantic characterization of the element at the word position, and a value for the feature, depending on the word position and the type of characterization. We describe these features in more detail below.

PDEP includes three corpora, collectively called the TPP Corpora (Litkowski, 2013a). The first was all FrameNet sentences (57 prepositions, 26739 instances), not just those used in SemEval (24 prepositions, which were divided into training and test sets). The second was a set of 20 sentences drawn from the Oxford English Corpus (OEC) to exemplify each sense in ODE, notably providing instances for multiword prepositional phrases (7485 sentences). The third was a set of sentences from the written portion of the British National Corpus, drawn with methodology used in the Corpus Pattern Analysis (CPA) project, typically with 250 instances for each preposition (47258 sentences). The CPA corpus was used as the basis for training an SVM model, with the SemEval and OEC corpora used as test sets.

1.1. Overview of Features in Preposition Disambiguation

On average, about 1250 features are generated for each sentence. In general, where there are 250 instances in the CPA corpus, this means that about 300,000 features are generated. For this general case, about 70,000 distinct features are generated, so that each distinct feature value occurs about 4 times. Of course, many features have a much higher frequency, perhaps occurring in nearly all the instances, while others occur just one time. The SVM modeling takes these into account, generating a weight for each distinct feature. In Litkowski (2016), using 10-fold cross-validation, we showed that the CPA-based models were internally consistent, achieving an accuracy of 80 percent. This suggests that this corpus is generally representative of preposition behavior.

We applied the models to the two test sets and obtained much lower accuracies, 46 percent for the SemEval corpus and 49 percent for the OEC corpus. These results suggest that these two corpora are not representative. Notwithstanding, we used these results as the basis for explorations that might lead to improvements. One question concerned the relative importance and redundancy of the features. In Litkowski (2016), we described a process of recursive feature elimination (RFE) following the method described in Guyon et al. (2002).

RFE starts with the full set of features, trains the SVM model, and examines the weights of the features. At each stage, half of the features are eliminated, except for the first iteration, where the number of features eliminated takes the number to the power of two less than the total number. The features eliminated at each stage are those with the lowest squared coefficients in the SVM model. The process is continued until only one feature remains. At each stage, a new SVM model is trained and applied to the test sets. A record is kept of the accuracies at each feature number. We then examined this data to identify the lowest number of features that produced the highest accuracies. Overall, these optimum levels improved the accuracies on the two test sets by 4 or 5 percent, and with a reduction in the number of features needed by 90 percent. Thus, improved performance was achieved with a fraction of the features, suggesting considerable redundancy among the features. This finding provides the motivation for an in-depth examination of the disambiguation features.

1.2. Characteristics of Features

As mentioned above, features consist of three components: (1) the word-finding rule, (2) the feature extraction rule, and (3) the feature value. The feature generation for an instance iterates through the two sets of rule types and identifies the value(s) to be associated with each combination. These are printed to a file consisting of an instance identifier, the assigned sense for that instance, and all of the features, with a separator between each element in the list (‘\30’). When examining a feature file, the separator facilitates splitting the features into distinct strings. Each word-finding rule and each feature extraction rule is specified with a one-, two-, or three-letter prefix. The feature value is a larger string.

1.2.1. Word-Finding Rules

In the Tratz-Hovy system, there are seven word-finding rules. The word-finding rules fall into two groups: words pertaining to the governor and words pertaining to the complement. The five governor word-finding rules are

- governor (**h**): the governing token of the preposition in the dependency parse
- verb or head to the left (**l**): the first token with a lower index in the dependency parse that has a verb part of speech or a noun, pronoun, or adjective label
- head to the left (**hl**): the first token with a lower index in the dependency parse that has a noun, verb, pronoun, or adjective label and which is identified as the head of the prepositional phrases
- verb to the left (**vl**): the first token to the left that has a verb label, and
- word to the left (**wl**): the first token to the left, if the preposition is not the first word.

As can be seen from the descriptions, there appears to be considerable similarity in the tokens that are identified with these rules. Such similarity can be a source of redundancy for the features that are generated as a result. However, each rule yields small differences in the sets of tokens that are identified; such differences will be highlighted in the feature ablation.

The two complement word-finding rules are

- syntactic preposition complement (**c**): views the preposition as a head in the dependency parse, and examines its children for tokens identified as preposition objects or complements, and
- heuristic preposition complement (**hr**): examines tokens that follow the preposition, looking at the part of speech to identify the most likely furthest complement (examining nouns, adjectives, pronouns, gerunds, and some specific words such as “some” or “each”).

Although the two methods for finding complements are somewhat different, they will generally find the same token. Again, some differences are expected.

1.2.2. **Feature Extraction Rules**

There are 17 feature extraction rules designed to characterize the tokens identified by the word-finding rules, i.e., to generate feature values. Some of these rules generate only a single value, while others generate a large number of values; we will describe the range of values and the number of occurrences compared to the number of instances for each rule. We have extended the set of rules used in the Tratz-Hovy system to include others features that might be of interest. The rules considered in this analysis are:

- word (**w**): the token itself (generally equal to the number of instances, with some occurring multiple times),
- lemma (**l**): the lemma for the token, if identifiable (generally equal to the number of instances, but with more duplicated values than the **w** values),
- word class (**wc**): one of four values, *noun*, *verb*, *adjective*, or *adverb* (generally slightly fewer than the number of instances),
- part of speech (**pos**): one of 37 values (generally only about half of these occur for a given preposition, but usually covering all instances),
- WordNet lexical name (**ln**): one of 40 values (with many occurring for a particular token, with several occurrences for each token, reflecting WordNet polysemy),

- WordNet immediate synonyms (**s**): the lexemes in the WordNet synsets for the token (a large number of possible values, with perhaps as many as 15 occurrences for each instance for each token)
- WordNet immediate hypernym (**h**): the lemmas in the immediate WordNet hypernyms (a large number of values, with perhaps 10 values for each token)
- all WordNet synonyms (**as**): extends the immediate WordNet synsets to include all derived forms and morphological variants, with three times the number of values and occurrences,
- all WordNet hypernyms (**h**): the lemmas in all WordNet hypernyms, up to 15 levels in the WordNet hierarchy (triples the numbers for WordNet immediate hypernyms)
- all WordNet gloss words (**g**): all words in all glosses of all the senses in WordNet of the token (by far, the largest set of features, as much as half of all features),
- whether the word is capitalized (**c**): the single value *true* when the token is capitalized (generally a low frequency feature),
- FrameNet entry (**fn**): a feature generated only when the head (**h**) is in the FrameNet dictionary and has the preposition as a frame element realization (a low frequency feature, occurring only for the more common prepositions),
- VerbNet entry (**vn**): a feature generated only when the head (**h**) is in the VerbNet dictionary and has the preposition as part of its specification (a low frequency feature, occurring only for the more common prepositions),
- verb from the pattern dictionary of English verbs (**cpa**): a feature generated only when the head (**h**) is in the pattern dictionary of English verbs and has the preposition as part of its specification, as described in Baisa et al. (2015) (a low frequency feature, occurring only for the more common prepositions),
- Oxford noun hierarchy immediate hypernym (**o**): a feature generated only for preposition complements (**hr**) and governors (**h**) that are nouns, where the noun is accessed in the Oxford Dictionary of English noun hierarchy to identify its immediate hypernyms, as described in McCracken (2004) (a moderately frequent feature, with the possibility of multiple hypernyms for a token),
- rule itself (**ri**): a feature with the sole value *rulefired* added when a word-finding rule is successful in finding a token (generally succeeds for all word-finding rule and for all instances, but there are usually many exceptions), and
- affixes (**af**): a feature that characterizes prefixes and suffixes present in the token (such as numerical prefixes and disease suffixes), (there are 27 possible affixes that are checked; they occur relatively frequently).

As can be seen, the feature values for the feature extraction rules are of many types, ranging from a small set of values to a large set of values, encoded in some cases and corresponding to ordinary words in other cases.

2. The Feature Ablation Algorithm

In general, we follow the procedures for feature ablation described in Fraser et al. (2014) and Bethard (2008), i.e., establishing and systematically removing feature sets to identify the most important features. This approach may be compared to many investigations attempting to determine the importance of features, i.e., leaving one out (LOO) and one-only (OO); This was used by Tratz (2011), in considering the features described above. Such a strategy essentially bookends the algorithm used here, with LOO being the first iteration and OO corresponding to the last iteration. We will discuss our results with those of Tratz.

The general procedure is shown in Algorithm 1. Unlike the two cited studies, we have three types of feature sets to examine: word-finding rules (**wfr**s), feature extraction rules (**fer**s), and WFR-FER combinations (**wfr:fer**s). This means we have 7 sets of word-finding rules, 17 sets of feature extraction rules, and up to 117 sets of combinations. Note, however, that not all combinations occur; for example, FrameNet feature extraction rules occur only with the governor word-finding rule. As a result, the number of combination sets is usually about 90 to 95.

Algorithm 1 Feature Ablation

```
Input: Preposition feature file and type
Output: Most important feature sets with
   statistics on effect of elimination
 1: count features from training set
 2: compute reference accuracy and reference
        feature set (R) using all features
 3: $n$ = number of feature sets for type
 4: compute base accuracy for each feature set
        of type by itself
 5: while $n \geq 2$ do
 6:   eliminate each remaining feature set from
        R
 7:   create SVM file and train SVM model
 8:   apply SVM model to test sets
 9:   s = identSetToRemove
 10. R = R – {features in s}, recording s
 11:   $n = n - 1$
 12: end while
```

This algorithm entails a very large number of SVM models for each type. One such model is generated in step 2. In step 4, an SVM model is generated for each feature set on its own, establishing a base accuracy that may be used in identifying a feature set to remove; this corresponds to the only-one strategy (OO). Step 7 is performed [(n+1) * n / 2]-1 times, when is the number of feature sets for the type. This is 27 times for word-finding rules, 152 times for feature extraction rules, and up to 6902 times for the combination feature sets (but usually about 4560 times when only 95 combinations occur in the data).

The SVM model in each calculation uses the CPA instances and only the features specified for that run. The amount of time for each model is a function of the number of instances, the number of features, the number of feature values, and the number of instances in the test set. Typically, with 250 instances, it takes about 2 hours for each preposition. For the 9 prepositions with 500 instances, it takes 6 hours; for the 4 prepositions with 750 instances, it takes 9 hours.

Each SVM stage generates several statistics. The first set is the number of correct and incorrect instances of the model when applied to the two test sets. These are used to calculate the accuracy of the model (used later in determining the feature set to be removed). The calculation also generates the average score of the classification model over all instances. In scoring, we combine both the SemEval and the OEC test sets, even though, as shown in Litkowski (2013b), the two sets are not from the same population. In addition, for about half of the prepositions, particularly the phrasal ones, there are no SemEval instances. Finally, each line identifies the number of distinct features used in the model and the number of occurrences of these features in the training set.

We provide additional information on several of the steps of the algorithm in the following sections.

2.1. Step 1: Counting Features in the Training Set

In this step, we read the feature file to establish the number of instances that are involved in training the SVM models. This particularly involves excluding instances for which feature generation did not succeed, instances tagged with a "pv" sense (indicating that the preposition is actually a part of a phrasal verb), and instances tagged with an "x" sense (indicating that the instance is ill-formed, e.g., actually an adverb or with some other problem). This step also establishes the feature dictionary, generating a frequency count of the features The keys in this dictionary constitute the initial reference set of features for later use in keeping track of which features are still under consideration as feature sets are removed.

2.2. Step 4: Determining the Base Accuracy of Each Feature Set

An array is established to keep track of the order in which feature sets are removed. We next determine the accuracy (and other statistics) for each feature set by itself. In doing so, we remove all other feature sets from the reference set of features. During this process, it may happen that the specific feature set may have no occurrences in the features for that preposition. Mostly, this occurs with the **wfr:fer** combinations, but may occur with individual word-finding or feature extraction rules. In these cases, such empty feature sets are added at the end of the removal array; there is no ordering among the feature sets that have no features in the instances.

The base accuracies for the feature sets may be used later in selecting which feature set to remove. It is worth noting that many feature sets have few features and few occurrences, but still yield an SVM model.

2.3. Steps 6 - 8: Removing and Scoring Feature Sets

Each iteration through these steps involves *n* assessments, each involving the removal of a feature set from the reference set of features. As *n* decreases, the size of the reference set steadily decreases as well. For each of the *n* remaining sets, the instant set is removed from the reference set to determine a set that is to be kept. This kept set is then scored against the test sets. The results for the *n* sets are put into an array for testing in the next step.

2.4. Steps 9: Identifying the Set to be Removed

This step implements the essence of the methods described in Fraser et al. (2014) and Bethard (2008) to identify the feature set to be removed. Essentially, this set is the one that has the least effect on accuracy. We iterate through the sets being evaluated (i.e., the array from the previous step). We keep track of the minimum difference from the reference accuracy (initially set to +∞) and the item index of the minimum set in the array. Several tests may be involved.

The first comparison is made to the reference accuracy, i.e., the one that uses all the features in its SVM model. We subtract the accuracy of the current item from the reference accuracy. We compare this to minimum difference that has been found so far. If the difference for the current item is greater than the current minimum, the current item is more important, so we continue to the next item. If the difference is less than the current minimum, the current item becomes the item to be removed and we continue to the next item. (On the first iteration, the difference will be less than +∞, so the first item will be the initial selection to be remove.) In many cases, however, the difference will be neither greater nor less than the reference accuracy, particularly when the number of features that have been removed is relatively small compared to the full feature set. This will often be the case when the initial reductions are being considered. Also, it may be the case that the difference from the reference accuracy is negative, i.e., removal of a feature set actually improves the accuracy. This was noted in Fraser et al., and occurs frequently in the examination of the feature sets in this investigation.

If the difference for the current item is the same as that for the current minimum, we next turn to a comparison of the base accuracies. If the base accuracy of the current item is less than that of the current minimum, the current item is selected as the item to be removed. If even these are the same, we finally compare the average classification scores and pick the item which has the lower score; since these scores are computed using the coefficients of the SVM model, they are almost assuredly different for the items being tested.

2.5. Steps 10: Removing a Feature Set from the Reference Set

The last step in the inner loop of the algorithm is removing the least important feature set denitrified in the last step. This decreases the size of the reference feature set, i.e., the features that are still active in the SVM modeling. The feature set is added to the array that holds the removal order of the feature sets. When the loop reaches the final iteration, i.e., there are just two feature sets remaining, removal of the less important one leaves just one as the most important. Note that when there are only two remaining, removal of one leaves just the base features for the other.

3. Results from Feature Set Ablation

Perhaps not surprisingly, the most important feature sets vary considerably, essentially different for each preposition. Even for the smallest set, the seven word-finding rules, there is considerable variation. This makes it difficult to interpret the results.

3.1. General Results

We performed feature ablation for 116 polysemous prepositions. This involved 25192 instances containing over 31 million features, an average of 1255 features per instance, ranging from 697 to 1771, with a standard deviation of 197. There are 8 million distinct features in the 116 sets, a frequency of 4 occurrences per feature. Many feature values occur only once, while others may occur for almost all the instances of a preposition. These feature counts do not represent the total number of unique feature values. For example, the noun word class of the preposition complement (**hr:wc**) is likely to occur for all prepositions. The number of unique feature values across all prepositions is likely to be considerably smaller.

The feature ablations for these prepositions involved the calculation of about 537,000 SVMS over a span of four weeks, an average of about 4630 ablations per preposition. On average, the ablations began with 70,000 features, ranging from 2401 (for *nigh* with 6 instances) to 163576 (for *of* with 713 instances). Each of the SVMs was evaluated against the two test sets, consisting of 33014 instances; the accuracies computed in these evaluations were used as the basis for making decisions as to which feature set was to be ablated at each step.

In general, each preposition is unique in terms of its most important feature sets. With over 5000 permutations of the seven word-finding rules, the chance of any two being the same is very small. For the 17 feature extraction rules and 119 combinations, the likelihood of similar orderings is even smaller. As a result, we can only look at general trends (primarily the average rank of the feature sets) and use the rankings as a guide to more detailed examination of the features.

3.2. Word-Finding Rule Ablation

Table 1 shows the average ranks and the proportion of word-finding rule features in the instance files used to train the SVMs. The table also shows the effect of leaving each set out (LOO) and of using only the set (OO) in using the SVMs. The average ranks, with values from 1 to 7, show a narrow range, suggesting that there is relatively little difference in the importance of the different rules. However, the two complement rules are on average ranked higher than the other rules, suggesting that these are slightly more important in disambiguating the prepositions. The next two rules in importance suggest that paying attention to the token immediately to the left of the preposition and also being able to identify the governor of the prepositional phrases are somewhat more important than the remaining rules.

Table 1 Word Finding Rule Ablation[1]

Word Finding Rule	LOO	OO	Rank	Proportion
Heuristic Complement (**hr**)	0.470	0.388	3.34	0.098
Syntactic Complement (**c**)	0.472	0.374	3.56	0.090
Word Left (**wl**)	0.473	0.415	3.91	0.103
Governor (**h**)	0.467	0.431	3.96	0.164
Head Left (**hl**)	0.474	0.417	4.05	0.126
Head or Verb Left (**l**)	0.478	0.417	4.33	0.235
Verb Left (vl)	0.480	0.337	4.85	0.184

It is also worth noting that each of the complement rules accounts for only about 10 percent of the features, again suggesting that a focus on these elements is relatively efficient in the disambiguation. This is also the case with using the token immediately to the left of the preposition. The LOO and OO differ from those of Tratz (2011). Here, we suggest that identifying the governor is the most important feature

[1] For comparison, the accuracy using all features is 0.475 on 33014 sentences in the two test sets. LOO is leave-one-out, showing the effect of leaving out the designated feature set. OO is one-only, showing the effect of using only the designated feature set.

on its own, while the complement rules are relatively less important on their own (consistent with Tratz). However, removing the complement features appears to lead to a somewhat larger degradation of performance. We suggest that the average ranks may provide a more accurate picture of relative importance.

3.3. Feature Extraction Rule Ablation

Table 2 shows the average ranks and the proportion of feature-extraction rule features in the instance files used to train the SVMs. The table also shows the effect of leaving each set out (LOO) and of using only the set (OO) in using the SVMs. The range of ranks, with values from 1 to 17, is greater than for the word-finding rules. The most important feature is all hypernyms (**ah**), followed by the WordNet lexicographer file name (**ln**), the part of speech (**pos**), and all synonyms (**as**) (which includes WordNet synsets and derivations). The next five features, somewhat clustered, are the word (**w**), the lemma (**l**), the word class (**wc**), the immediate synonyms (**s**), and the immediate hypernym (**h**). The rule itself (**ri**), affixes (**af**), the gloss words (**g**), and capitalization (**c**) also form a cluster. In the final cluster are features that we have added to incorporate various lexical resources: the Oxford noun hierarchy (**o**), FrameNet (**fn**), VerbNet (**vn**), and verbs from the pattern dictionary of English verbs (**cpa**).

The LOO and OO results are somewhat different from those of Tratz (2011). Tratz only discussed these results to a small extent. While his and our results showed a major effect for all hypernyms, our results are different for WordNet glosses, where he found a major effect. The glosses show an ability on their own (OO) that is second only to the hypernyms, but leaving it out appears to have a negative performance on overall accuracy. For the most part, the LOO results have only a small effect on performance, so it is difficult to distinguish much about the relative performance of the different rules. However, the WordNet lexicographer file names and the part of speech show more significance than shown by Tratz. Curiously, the reduction in accuracy for the added lexical resources (FrameNet, VerbNet, and the pattern dictionary of English verbs) was quite considerable in the LOO results, despite the fact that these resources performed quite poorly on their own (i.e., their OO results).

It is worth noting that rules which generate a large number of features are negatively correlated (-0.187) with the average ranks. For example, all hypernyms and all synonyms account for a large proportion of all features, 19 percent and 16 percent, respectively. In the case of all hypernyms, virtually any noun or verb is likely to lead all the way up the WordNet hierarchy. It is difficult to see how such features can be effective discriminators or predictors of the sense. It is possible that the significance of these features lies at some intermediate levels of the hierarchy. To some extent, this hypothesis is supported by the second-highest feature set, the WordNet lexicographer file name, which provides a hint of semantic characterization of the features. We explore this further when considering the **wfr:fer** combinations.

The fact that features generated by the four lexical resources may not be reflective of their significance. In the case of FrameNet, VerbNet, and the pattern dictionary of English verbs, very few features are generated. This is due in part to that these features were generated only for the governors of the prepositional phrase. This may also be due in part to the recognized lack of coverage of these resources.

3.4. Word-Finding Feature-Extraction Rule Combination Ablation

Table 3 shows the average ranks and the proportion of word-finding feature-extraction rule combination features in the instance files used to train the SVMs. Since the proportions can be very small, the table also shows the count of each combination. The table also shows the effect of leaving each set out (LOO) and of using only the set (OO) in using the SVMs. The range of ranks, with values from 1 to 119, is greater than for the word-finding rules, with average ranks ranging from 30 to 119.

The most important feature sets identified in this table bear a similarity to those shown as most important in the feature extraction rule ablation (Table 2). Five of the six most important feature sets involve the all hypernyms feature (**ah**), i.e., for five of the seven word-finding rules. This same feature is also ranked highly for the other two word-finding rules (13th and 24th). The next two most important feature extraction rule (the WordNet lexicographer file name, **ln**, and the part of speech, **pos**) begin to appear in average ranks below the all hypernym ranks. Thus, the combination results appear to support the results from the feature extraction rule ablation.

Further examination of the combination results suggests some clustering in average ranks for each of the other feature extraction rules. This again seems to support the average ranks for those rules.

To some extent, there appears to a negative correlation (-0.36) between the average ranks and the frequency counts for the features. The top-ranked combination features seem to occur more often, although not so much the case for file names and parts of speech features. This would suggest that these latter features may be of more significance than suggested by their ranks; the relative importance of the WordNet file names is consistent with the notion of semantic word sketches as suggested by McCarthy et al. (2015). Many of the top-ranked features also correspond to having multiple values for the token that gives rise to the features. Thus, a given token can generate multiple hypernym values, but only one part of speech, word, or lemma. Several of the features occur with a much lower frequency than the total number of instances. For example, capitalization of a token will occur much less frequently than lowercase forms.

The LOO and OO results for the combination feature sets do not seem to provide any further insights into these feature sets. Tratz (2011) did not investigate these combinations. Our LOO results show little variation, changing the accuracy only by a few tenths of a percent in each case. The OO results show more variation, with drops in accuracy from 10 to 20 percentage points.

In the discussion of feature extraction rules for the additional lexical resources, we indicated that these are constrained considerably by the coverage of these resources. For the combination features, as a result, they appear at the bottom of the average ranks. In many cases, no features were generated (by design), but even when they occur, their frequency is relatively small and thus appearing as not very important.

4. Interpreting the Results

We have performed feature set ablations for 116 prepositions. Our results suggest that each preposition has its own array of important feature sets. This variation makes it difficult to attach much significance to the results for individual prepositions. As a result, we computed the average ranks for the three types of feature sets. These averages permitted a more general interpretation of the significance of the feature sets in each type. However, at the same time, these averages showed a negative correlation with the number of features in each set, suggesting that the sheer number of features may have a disproportionate effect on which feature sets are deemed most important.

In general, features characterizing the preposition complement are slightly more important than features characterizing the context. Among the contextual features, the token to the left is most important, followed by those characterizing the governor. Feature sets characterizing the semantics of tokens, all hypernyms and WordNet lexicographer file names, were generally most important among feature extraction rules, followed by the part of speech, characterizing the syntax for the tokens. The relative importance of these features was shown in the ablations for feature extraction rules, as well as the combination sets.

These observations are of a qualitative type usually found in ablation studies. However, such studies usually are concerned with a much smaller number of feature sets. In particular, the results here do not seem to provide information that can be added directly to describe the behavior of individual prepositions. Instead, these results suggest the need for further study in two directions. First, we can attempt more detailed analysis of selected features, looking at individual feature values and determining their relative importance; however, for some feature types, such as all hypernyms, targeting which features to examine may be difficult. Second, we can examine the ablation results in conjunction with the results of the recursive feature elimination studies as described in Litkowski (2016). Initial exploration of these data does not, however, suggest a clear path forward.

References

Vit Baisa, Jane Bradbury, Silvie Cinkova, Ismail El Maarouf, Adam Kilgarriff, and Octavian Popescu. 2015. SemEval-2015: Task 15: A Corpus Pattern Analysis Dictionary-Entry-Building Task. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015)*. Denver, Colorado, USA, 315-24.

Steven Bethard. 2008. *Finding Event, Temporal and Causal Structure in Text: A Machine Learning Approach*. PhD Thesis, University of Colorado.

Kathleen Fraser, Graeme Hirst, Naida Graham, Jed Meltzer, Sandra Black, and Elizabeth Roahon. 2014. Comparison of different feature sets for identification of variants in progressive aphasia. In *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*. Baltimore, Maryland, USA, 17-26.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Valdimir Vapnik. 2002. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46: 389-422.

Ken Litkowski. 2016. Pattern Dictionary of English Prepositions. In M. Diab, A. Villavicencio, M. Apidianaki, V. Kordoni, A. Korhonen, P. Nakov, and M. Stevenson, editors *Essays in Lexical Semantics and Computational Lexicography – In Honor of Adam Kilgarriff*. Springer Series Text, Speech, and Language Technology. Springer. In press.

Ken Litkowski. 2014. Pattern Dictionary of English Prepositions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, Maryland, USA, pp. 1274-83.

Ken Litkowski. 2013a. *The Preposition Project Corpora*. Technical Report 13-01. Damascus, MD: CL Research.

Ken Litkowski. 2013b. *Preposition Disambiguation: Still a Problem*. Technical Report 13-02. Damascus, MD: CL Research.

Diana McCarthy, Adam Kilgarriff, Milos Jakubicek, and Siva Reddy.2015. Semantic Word Sketches. *Corpus Linguistics 2015*. Lancaster University.

James McCracken. 2004. Rebuilding the Oxford Dictionary of English as a Semantic Network. In: *COLING 2004 Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*. Geneva, Switzerland, pp. 69-72.

Stephen Tratz. 2011. *Semantically-Enriched Parsing for Natural Language Understanding*. PhD Thesis, University of Southern California.

Stephen Tratz and Eduard Hovy. 2011. A Fast, Accurate, Non-Projective, Semantically-Enriched Parser. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.

Table 2 Feature Extraction Rule Ablation[2]

Feature-Extraction Rule	LOO	OO	Rank	Proportion
All hypernyms (**ah**)	0.453	0.465	5.84	0.19389
Lexicographer file name (**ln**)	0.472	0.425	5.91	0.01849
Part of speech (**pos**)	0.472	0.325	6.33	0.00572
All synonyms (**as**)	0.478	0.376	7.10	0.15572
Word (**w**)	0.475	0.277	7.22	0.00572
Lemma (**l**)	0.475	0.312	7.30	0.00550
Word class (**wc**)	0.474	0.304	7.39	0.00540
Synonyms (**s**)	0.477	0.362	7.48	0.06042
Immediate hypernym (**h**)	0.476	0.392	7.83	0.07744
Rule itself (**ri**)	0.476	0.267	8.46	0.00529
Affixes (**af**)	0.476	0.294	8.76	0.00671
Gloss words (**g**)	0.481	0.414	9.26	0.45475
Capitalization (**c**)	0.475	0.159	9.59	0.00061
Oxford hypernym (**o**)	0.475	0.301	9.91	0.00429
FrameNet (**fn**)	0.468	0.193	13.94	0.00005
VerbNet (**vn**)	0.453	0.094	14.93	0.00001
Corpus pattern verb (cpa)	0.450	0.081	15.74	0.00001

[2] For comparison, the accuracy using all features is 0.475 on 33014 sentences in the two test sets. LOO is leave-one-out, showing the effect of leaving out the designated feature set. OO is one-only, showing the effect of using only the designated feature set.

Table 3 WFR:FER Combination Feature Ablation[3]

Feature	LOO	OO	Rank	Proportion	Count
hl:ah:	0.474	0.381	29.88	0.02530	800233
hr:ah:	0.474	0.339	31.95	0.02567	811922
h:ah:	0.472	0.397	32.96	0.02816	890508
hr:g:	0.477	0.320	33.46	0.03974	1256879
c:ah:	0.473	0.331	33.72	0.02430	768403
l:ah:	0.474	0.398	33.91	0.04275	1352098
hr:pos:	0.475	0.282	34.17	0.00078	24824
l:ln:	0.474	0.360	34.50	0.00429	135712
hr:ln:	0.476	0.306	35.33	0.00202	63730
wl:pos:	0.475	0.289	35.73	0.00076	24120
hr:h:	0.476	0.280	36.66	0.00683	215870
wl:ah:	0.474	0.360	36.78	0.01984	627342
hl:pos:	0.476	0.299	36.94	0.00075	23846
h:ln:	0.475	0.361	36.98	0.00291	91930
c:ln:	0.475	0.302	37.50	0.00191	60252
hl:g:	0.476	0.355	37.53	0.05784	1829120
c:h:	0.475	0.273	37.72	0.00644	203592
hl:ln:	0.475	0.357	38.03	0.00237	75100
l:h:	0.476	0.348	38.08	0.01902	601425
l:as:	0.477	0.333	38.53	0.03996	1263745
h:as:	0.475	0.324	38.59	0.02735	864952
l:pos:	0.475	0.296	38.64	0.00118	37406
vl:ah:	0.475	0.319	38.69	0.02786	881218
h:pos:	0.476	0.287	39.03	0.00079	25031
c:pos:	0.475	0.272	39.05	0.00074	23505
l:g:	0.477	0.369	39.20	0.10764	3404297
h:h:	0.475	0.343	39.87	0.01291	408343
l:s:	0.476	0.306	40.86	0.01482	468549
l:wc:	0.475	0.275	41.22	0.00117	36874
hl:as:	0.476	0.299	41.76	0.01895	599285
hl:h:	0.476	0.323	41.84	0.00954	301743
h:g:	0.476	0.373	42.00	0.07623	2410936
hl:wc:	0.475	0.280	42.01	0.00074	23314
vl:h:	0.476	0.301	42.07	0.01521	481122
c:as:	0.475	0.240	42.14	0.01001	316542
h:ri:	0.475	0.272	42.33	0.00079	25031
vl:ln:	0.475	0.296	42.54	0.00308	97292
vl:g:	0.478	0.312	43.11	0.08758	2769721
c:g:	0.477	0.306	43.12	0.03779	1195219

[3] For comparison, the accuracy using all features is 0.475 on 33014 sentences in the two test sets. LOO is leave-one-out, showing the effect of leaving out the designated feature set. OO is one-only, showing the effect of using only the designated feature set.

Feature	LOO	OO	Rank	Proportion	Count
hr:as:	0.475	0.244	43.22	0.01059	334952
hl:ri:	0.476	0.272	44.09	0.00075	23846
l:ri:	0.476	0.263	44.28	0.00075	23846
vl:pos:	0.476	0.259	45.21	0.00070	22082
hr:ri:	0.475	0.272	45.30	0.00078	24824
wl:ri:	0.476	0.263	45.46	0.00076	24120
h:wc:	0.475	0.282	45.91	0.00075	23785
wl:as:	0.476	0.288	46.04	0.01513	478578
wl:g:	0.476	0.342	46.28	0.04792	1515527
vl:as:	0.476	0.279	46.70	0.03372	1066490
wl:wc:	0.475	0.291	46.78	0.00062	19651
wl:ln:	0.475	0.340	46.87	0.00192	60673
hr:l:	0.475	0.213	47.78	0.00073	23185
hr:w:	0.475	0.201	47.97	0.00078	24824
vl:s:	0.476	0.265	48.48	0.01172	370731
h:s:	0.475	0.286	48.61	0.01003	317170
vl:ri:	0.475	0.251	48.66	0.00070	22082
wl:h:	0.476	0.301	48.79	0.00749	237009
hr:wc:	0.475	0.258	48.83	0.00073	23124
c:s:	0.475	0.222	49.01	0.00490	155062
hl:s:	0.475	0.256	49.06	0.00757	239292
c:ri:	0.475	0.262	49.08	0.00074	23477
vl:wc:	0.475	0.251	49.32	0.00070	22082
hr:s:	0.476	0.226	49.66	0.00518	163778
l:af:	0.475	0.260	50.28	0.00138	43568
l:l:	0.476	0.236	50.35	0.00115	36252
h:af:	0.475	0.250	50.88	0.00095	30180
c:l:	0.475	0.209	51.41	0.00070	22040
c:wc:	0.475	0.250	51.81	0.00069	21827
wl:s:	0.475	0.248	52.06	0.00620	196198
l:w:	0.475	0.195	52.32	0.00118	37404
c:af:	0.475	0.201	52.34	0.00086	27274
vl:af:	0.475	0.219	52.71	0.00086	27284
wl:l:	0.475	0.190	52.91	0.00074	23374
hl:af:	0.475	0.245	53.98	0.00089	28217
wl:af:	0.475	0.236	54.24	0.00083	26305
hr:af:	0.476	0.208	54.42	0.00093	29263
h:l:	0.475	0.206	54.44	0.00077	24334
vl:l:	0.475	0.202	55.42	0.00069	21945
c:w:	0.475	0.198	55.76	0.00074	23508
wl:w:	0.475	0.170	57.12	0.00076	24120
vl:w:	0.475	0.167	57.84	0.00070	22082
hl:w:	0.475	0.166	58.72	0.00075	23846
h:w:	0.475	0.169	59.31	0.00079	25031

Feature	LOO	OO	Rank	Proportion	Count
hr:o:	0.476	0.253	59.85	0.00315	99695
hl:l:	0.475	0.189	60.73	0.00072	22778
wl:c:	0.475	0.119	66.72	0.00014	4419
hr:c:	0.476	0.146	69.73	0.00013	4222
c:c:	0.475	0.144	70.77	0.00012	3866
h:o:	0.475	0.195	71.16	0.00114	36017
l:c:	0.475	0.116	71.98	0.00009	2728
hl:c:	0.475	0.114	72.47	0.00008	2561
h:c:	0.474	0.110	78.61	0.00004	1177
vl:c:	0.468	0.103	81.07	0.00001	340
wl:vn:	--	--	93.94	0.00000	0
h:fn:	0.468	0.194	94.13	0.00005	1693
wl:o:	--	--	94.95	0.00000	0
wl:fn:	--	--	95.97	0.00000	0
wl:cpa:	--	--	96.97	0.00000	0
vl:vn:	--	--	98.03	0.00000	0
vl:o:	--	--	99.03	0.00000	0
vl:fn:	--	--	100.03	0.00000	0
vl:cpa:	--	--	101.03	0.00000	0
l:vn:	--	--	102.28	0.00000	0
l:o:	--	--	103.28	0.00000	0
l:fn:	--	--	104.28	0.00000	0
l:cpa:	--	--	105.28	0.00000	0
hr:vn:	--	--	106.32	0.00000	0
h:cpa:	0.450	0.083	106.64	0.00001	161
hr:fn:	--	--	107.32	0.00000	0
h:vn:	0.453	0.094	108.22	0.00001	371
hr:cpa:	--	--	108.32	0.00000	0
hl:vn:	--	--	109.37	0.00000	0
hl:o:	--	--	110.37	0.00000	0
hl:fn:	--	--	111.37	0.00000	0
hl:cpa:	--	--	112.37	0.00000	0
c:vn:	--	--	115.74	0.00000	0
c:o:	--	--	116.79	0.00000	0
c:fn:	--	--	117.86	0.00000	0
c:cpa:	--	--	118.86	0.00000	0