Predicting Word Clipping with Latent Semantic Analysis

Julian Brooke **Tong Wang** **Graeme Hirst**
Department of Computer Science
University of Toronto
{jbrooke,tong,gh}@cs.toronto.edu

Abstract

In this paper, we compare a resource-driven approach with a task-specific classification model for a new near-synonym word choice sub-task, predicting whether a full or a clipped form of a word will be used (e.g. *doctor* or *doc*) in a given context. Our results indicate that the resource-driven approach, the use of a formality lexicon, can provide competitive performance, with the parameters of the task-specific model mirroring the parameters under which the lexicon was built.

1 Introduction

Lexical resources, though the focus of much work in computational linguistics, often compare poorly to direct statistical methods when applied to problems such as sentiment classification (Kennedy and Inkpen, 2006). Nevertheless, such resources offer advantages in terms of human interpretability and portability to many different tasks (Argamon et al., 2007). In this paper, we introduce a new sub-task of near-synonym word choice (Edmonds and Hirst, 2002), prediction of word clipping, in order to provide some new evidence that resource-driven approaches have general potential.

Clipping is a type of word formation where the beginning and/or the end of a longer word is omitted (Kreidler, 1979). This phenomenon is attested in various languages; well-known examples in English include words such as *hippo* (*hippopotamus*) and *blog* (*weblog*). Clipping and related kinds of word formation have received attention in computational linguistics with respect to the task of identifying source words from abbreviated forms, which has been studied, for instance, in the biomedical and text messaging domains (Okazaki and Ananiadou, 2006; Cook and Stevenson, 2009).

Compared to many near-synonyms, clipped forms have the important property that the differences between full and abbreviated forms are almost entirely connotational or stylistic, closely tied to the formality of the discourse.[1] This fact allows us to pursue two distinct though related approaches to this task, comparing a supervised model of word choice (Wang and Hirst, 2010) with a mostly unsupervised system that leverages an automatically-built lexicon of formality (Brooke et al., 2010). Our findings indicate that the lexicon-based method is highly competitive with the supervised, task-specific method. Both models approach the human performance evidenced in an independent crowdsourced annotation.

2 Methods

2.1 Latent Semantic Analysis

Both approaches that we are investigating make use of Latent Semantic Analysis (LSA) as a dimensionality-reduction technique (Landauer and Dumais, 1997).[2] In LSA, the first step is to create a matrix representing the association between words as determined by their co-occurrence in a corpus, and then apply singular value decomposition (SVD) to identify the first k most significant dimensions of variation. After this step, each word can be represented as a vector of length k, which can be compared or combined with the vectors of other words. The best k is usually determined empirically. For a more detailed introduction to this method, see also the discussion by Turney and Littman (2003).

[1] Shortened forms might also be preferred in cases where space is at a premium, e.g. newspaper headlines or tweets.

[2] Note that neither technique is feasible using the full co-occurrence vectors, which have several hundred thousand dimensions in both cases; in addition, previous work has shown that performance drops off with increased dimensionality.

2.2 Classifying Context Vectors

Our first method is the lexical choice model proposed by Wang and Hirst (2010). This approach performs SVD on a term–term co-occurrence matrix, which has been shown to outperform traditional LSA models that use term–document co-occurrence information. Specifically, a given word w is initially represented by a vector v of all its co-occurring words in a small collocation context (a 5-word window), i.e., $v = (v_1, \ldots, v_n)$, where n is the size of the vocabulary, and $v_i = 1$ if w co-occurs with the i-th word in lexicon, or $v_i = 0$ otherwise. The dimensionality of the original vector is then reduced by SVD.

A context, typically comprising a set of words within a small collocation context around the target word for prediction (though we test larger contexts here), is represented by a weighted centroid of the word vectors. Together with the candidate words for prediction, this context vector can then be used as a feature vector for supervised learning; we follow Wang and Hirst in using support vector machines (SVMs) as implemented in WEKA (Witten and Frank, 2005), training a separate classifier for each full/clipped word form pair. The prediction performance varies by k, which can be tested efficiently by simply truncating a single high-k vector to smaller dimensions. The optimal k value reported by Wang and Hirst testing on a standard set of seven near-synonyms was 415; they achieved an accuracy of 74.5%, an improvement over previous statistical approaches, e.g. Inkpen (2007).

2.3 Using Formality Lexicons

The competing method involves building lexicons of formality, using our method from Brooke et al. (2010), which is itself an adaption of an approach used for sentiment lexicon building (Turney and Littman, 2003). Though it relies on LSA, there are several key differences as compared to the context vector approach. First, the pre-LSA matrix is a binary word–document matrix, rather than word–word. For the LSA step, we showed that a very low k value (20) was appropriate choice for identifying variation in formality. After dimensionality reduction, each word vector is compared, using cosine similarity, to words from two sets of seed terms, each representing prototypical formal and informal words, which provides a formality score for each word in the range of -1 to 1. The derivation of the final formality score involves several normalization steps, and therefore a full discussion is precluded here for space reasons; for the details, please see Brooke et al. (2010). Our evaluation suggests that, given a large-enough blog corpus, this method almost perfectly distinguishes words of extreme formality, and is able to identify the more formal of two near-synonyms over 80% of the time, better than a word-length baseline.

Given a lexicon of formality scores, the preferred form for a context is identified by averaging the formality scores of the words in the context and comparing the average score to a cutoff value. Here, the context is generally understood to be the entire text, though we also test smaller contexts. We take the cutoff to be midpoints of the average scores for the contexts of known instances; although technically supervised, we have found that in practice just a few instances is enough to find a stable, high-performing cutoff. Note that the cutoff is analogous to the decision hyperplane of an SVM. In our case, building a lexical resource corresponds to additional task-independent reduction in the dimensionality of the space, greatly simplifying the decision.

3 Resources

Blog data is an ideal resource for this task, since it clearly contains a wide variety of language registers. For our exploration here, we used a collection of over 900,000 blogs (216 million tokens) originally crawled from the web in May 2008. We segmented the texts, filtered out short documents (less than 100 words), and then split the corpus into two halves, training and testing. For each of the two methods described in the previous section, we derived the corresponding LSA-reduced vectors for all lower-case words using the collocation information contained within the training portion.[3] The testing portion was used only as a source for test contexts.

We independently collected a set of common full/clipped word pairs from web resources such as Wikipedia, limiting ourselves to phonologically-realized clippings. This excludes orthographic shortenings like *thx* or *ppl* which cannot be pro-

[3] We used the same dataset for each method so that the difference in raw co-occurence information available to each method was not a confounding factor. However, we also tested the lexicon method using the full formality lexicon from Brooke et al. (2010), built on the larger ICWSM blog corpus; the difference in performance was negligible.

nounced. We also removed pairs where one of the words was quite rare (fewer than 150 tokens in the entire corpus) or where, based on examples pulled from the corpus, there was a common confounding homonym–for instance the word *prob*, which is a common clipped form of both *problem* and *probably*. However, we did keep words like *doc*, where the *doctor* sense was much more common than the *document* sense. After this filtering, 38 full/clipped word pairs remained in our set. For each pair, we automatically extracted a sample of usage contexts from texts in the corpus where only one of the two forms appears. For each word form in each of our training and testing corpora, we manually removed duplicate and near-duplicate contexts, non-English and unintelligible contexts, and any remaining instances of homonymy until we had 50 acceptable usage examples for each word form in each sub-corpus (100 for each of the word pairs), a total of 3800 contexts for each of training and testing.

One gold standard is provided by the original choice of the writer, but another possible comparison is with reference to an independent human annotation, as has been done for other near-synonym word choice test sets (Inkpen, 2007). For our annotation, we used the crowdsourcing website Crowdflower (www.crowdflower.com), which is built on top of the well-known Amazon Mechanical Turk (www.mturk.com), which has been used, for instance, to create emotion lexicons (Mohammad and Turney, 2010). In general terms, these crowdsourcing platforms provide access to a pool of online workers who do small tasks (HITs) for a few cents each. Crowdflower, in particular, offers a worker-filtering feature where gold standard HITs (75 clear instances taken from the training data) are interspersed within the test HITs, and workers are removed from the task if they fail to answer a certain percentage correct (90%). For each word form, we randomly selected 20 of 50 test contexts to be judged, or 1520 altogether. For each case, the workers were presented with the word pair and three sentences of context (additional context was provided if less than 40 tokens), and asked to guess which word the writer used. To get more information and allow participants to express a tentative opinion, we gave the workers five options for a word pair A/B: "Probably A/B", "Definitely A/B", and "I'm not sure"; for our purposes here, however, we will not distinguish between "Probably" and "Definitely". We queried for five different judgments per test case in our test corpus, and took the majority judgment as the standard, or "I'm not sure" if there was no majority judgment.

4 Evaluation

First, we compare our crowdsourced annotation to our writer's choice gold standard, which provides a useful baseline for the difficulty of the task. The agreement is surprisingly low; even if "I'm not sure" responses are discounted, agreement with the writer's choice gold standard is just 71.7% for the remaining datapoints. For certain words (such as *professor*, *doctor*), workers avoided the non-standard clipped forms almost entirely, though there were other pairs, like *photo/photograph*, where the clipped form dominated. Expected frequency, rather than document context, is clearly playing a role here.

Our main evaluation consists of comparing the predictions of our two methods to the original choice of the writer, as seen in our corpus. Accuracy is calculated as the number of predictions that agree with this standard across all the (3800) contexts in our test set. We first calibrated each model using the training set, and then prompted for predictions with various amount of context.[4] The 3-sentence context includes the sentences where the word appeared, and the sentences on either side. Other options we investigated were, for the vector classification, the option of using a single classifier for all pairs, or using a different k-value for each pair, and, for the lexicon-based prediction, the option of using a single cutoff for all pairs. The best k were determined by 10-fold cross-validation on the training set. The results are given in Table 1. Since our test sets are balanced, the random guessing performance is 50%.

A chi-square test indicates the difference between the best performing result for each method is not statistically significant. We see that both methods show an improvement with the addition of context beyond the sentence where the word appears, with full document context providing the best results; the improvement with full document context is statistically significant for the vector classification model ($p < 0.001$). Overall, the two methods make similar choices, with the agreement

[4] In all cases, other appearances of the word or an inflected form in the context were removed.

Table 1: Clipping prediction results, all pairs

Vector classification	
Options	**Accuracy**
Sentence context only (k=17)	62.9
3-sentence context (k=15)	64.6
Full document (FD) (k=16)	**67.9**
FD, single (generalized) classifier	66.8
FD, best k for each pair	65.9
Formality lexicon	
Options	**Accuracy**
Sentence context only	65.2
3-sentence context	65.5
Full document (FD)	**66.7**
FD, single (generalized) cutoff	65.1

Table 2: Clipping prediction results, by pair

	Accuracy	
Clipped pair	**VC model**	**FL model**
prof/professor	68	74
tourney/tournament	64	55
plane/airplane	61	42
doc/doctor	81	78
stats/statistics	74	75
meds/medication	82	82
fridge/refrigerator	65	63
app/application	66	62
mic/microphone	54	59
fam/family	84	85

of the predictions at 78.1% for the full document models. Another result that points to the similarity of the final models is that the best single k value is very close to the best k value for lexicon building from Brooke et al. (2010) The generalized clipping models (of both kinds) do worse than the pair-specific models, but the drop is fairly modest. An even more individualized vector classification model, in the form of individual k values for each pair, does not improve performance. If we instead take the worker judgements as a gold standard, the performance of our two models on that subset of the test data is worse than with a writer-based standard: 61.1% for the best lexicon-based model, and 63.6% for the best vector classification model.

Finally, we look at individual full/clipped word pairs. Table 2 contains the results for a sample of these pairs, using the best models from Table 1. Some word pairs (e.g. *mic/microphone*) were very difficult for the models, while others can usually be distinguished. The main difference between the two models is that there are certain pairs (e.g. *plane/airplane*) where the vector classification works much better, perhaps indicating that formality is not the most relevant kind of variation for these pairs.

5 Discussion

Our initial hypothesis was that the formality of the discourse plays a key role in determining whether a clipped form of a word will be used in place of a full form, and thus a lexicon of formality could be a useful tool for this kind of word choice. Our results mostly bear this out: although the vector classification model has a slight advantage, the lexicon-based method, which has the advantage of compactness, interpretability, and portability, does reasonably well. Tellingly, the best vector-based model is very similar to the lexicon in terms of its parameters, including a preference for the use of the entire document as context window and low LSA k, rather than the local context and high LSA k that was preferred for a previous near-synonym choice task (Wang and Hirst, 2010). In comparison to that task, clipping prediction is clearly more difficult, a fact that is confirmed by the results of our crowdsourced annotation.

The fact that the models do better on certain individual word pairs and more poorly on others indicates that the degree of formality difference between clipped and full forms is probably quite variable, and in some cases may be barely noticeable. Under those circumstances, the advantages of a vector classification model, which might base the classification on other kinds of relevant context (e.g. topic), are clear. We conclude by noting that for a highly specialized problem such as word clipping prediction, a single lexical resource can, it appears, complete with a task-based supervised approach, but even here we see signs that a single resource might be insufficient to cover all cases. For wider, more complex tasks, any particular resource may address only a limited part of the task space, and therefore a good deal of work may be required before a lexicon-based method can reasonably compete with a more straightforward statistical approach.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

References

Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. 2007. Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, 7:91–109.

Julian Brooke, Tong Wang, and Graeme Hirst. 2010. Automatic acquisition of lexical formality. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)*, pages 90–98, Beijing.

Paul Cook and Suzanne Stevenson. 2009. An unsupervised model for text message normalization. In *Proceedings of the NAACL HLT 2009 Workshop on Computational Approaches to Linguistic Creativity*, pages 71–79, Boulder, Colorado.

Philip Edmonds and Graeme Hirst. 2002. Near-synonymy and lexical choice. *Computational Linguistics*, 28(2):105–144, June.

Diana Inkpen. 2007. A statistical model for near-synonym choice. *ACM Transactions on Speech and Language Processing*, 4(1):1–17.

Alistair Kennedy and Diana Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22:110–1225.

Charles Kreidler. 1979. Creating new words by shortening. *Journal of English Linguistics*, 13:24–36.

Thomas K. Landauer and Susan Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Los Angeles.

Naoaki Okazaki and Sophia Ananiadou. 2006. A term recognition approach to acronym recognition. In *Proceedings of the Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL '06)*.

Peter Turney and Michael Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346.

Tong Wang and Graeme Hirst. 2010. Near-synonym lexical choice in latent semantic space. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)*, pages 1182–1190, Beijing.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco.