Distinguishing Common and Proper Nouns

Judita Preiss and **Mark Stevenson**
{j.preiss, r.m.stevenson}@sheffield.ac.uk
Department of Computer Science,
University of Sheffield
211 Portobello, Sheffield S1 4DP
United Kingdom

Abstract

We describe a number of techniques for automatically deriving lists of common and proper nouns, and show that the distinction between the two can be made automatically using a vector space model learning algorithm. We present a direct evaluation on the British National Corpus, and application based evaluations on Twitter messages and on automatic speech recognition (where the system could be employed to restore case).

1 Introduction

Some nouns are homographs (they have the same written form, but different meaning) which can be used to denote either a common or proper noun, for example the word *apple* in the following examples: (1) **Apple** designs and creates iPod (2) The **Apple** II series is a set of 8-bit home computers (3) The **apple** is the pomaceous fruit of the apple tree (4) For **apple** enthusiasts – tasting notes and apple identification.

The common and proper uses are not always as clearly distinct as in this example; for example, a specific instance of a common noun, e.g., *District Court* turns *court* into a proper noun.

While heuristically, proper nouns often start with a capital letter in English, capitalization can be inconsistent, incorrect or omitted, and the presence or absence of an article cannot be relied on.

The problem of distinguishing between common and proper usages of nouns has not received much attention within language processing, despite being an important component for many tasks including machine translation (Lopez, 2008; Hermjakob et al., 2008), sentiment analysis (Pang and Lee, 2008; Wilson et al., 2009) and topic tracking (Petrović et al., 2010). Approaches to the problem also have applications to tasks such as web search (Chen et al., 1998; Baeza-Yates and Ribeiro-Neto, 2011), and case restoration (e.g., in automatic speech recognition output) (Baldwin et al., 2009), but frequently involve the manual creation of a gazeteer (a list of proper nouns), which suffer not only from omissions but also often do not allow the listed words to assume their common role in text.

This paper presents methods for generating lists of nouns that have both common and proper usages (Section 2) and methods for identifying the type of usage (Section 3) which are evaluated using data derived automatically from the BNC (Section 4) and on two applications (Section 5). It shows that it is difficult to automatically construct lists of ambiguous nouns but also that they can be distinguished effectively using standard features from Word Sense Disambiguation.

2 Generating Lists of Nouns

To our knowledge, no comprehensive list of common nouns with proper noun usage is available. We develop a number of heuristics to generate such lists automatically.

Part of speech tags A number of part of speech (PoS) taggers assign different tags to common and proper nouns. Ambiguous nouns are identified by tagging a corpus and extracting those that have had both tags assigned, together with the frequency of occurrence of the common/proper usage. The CLAWS (Garside, 1987) and the RASP taggers

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*, pages 80–84, Atlanta, Georgia, June 13-14, 2013.

(Briscoe et al., 2006) were applied to the British National Corpus (BNC) (Leech, 1992) to generate the lists *BNCclaws* and *BNCrasp* respectively. In addition the RASP tagger was also run over the 1.75 billion word Gigaword corpus (Graff, 2003) to extract the list *Gigaword*.

Capitalization Nouns appearing intra-sententially with both lower and upper case first letters are assumed to be ambiguous. This technique is applied to the 5-grams from the Google corpus (Brants and Franz, 2006) and the BNC (creating the lists *5-grams* and *BNCcaps*).

Wikipedia includes disambiguation pages for ambiguous words which provide information about their potential usage. Wikipedia pages for nouns with senses (according to the disambiguation page) in a set of predefined categories were identified to form the list *Wikipedia*.

Named entity recognition The Stanford Named Entity Recogniser (Finkel et al., 2005) was run over the BNC and any nouns that occur in the corpus with both named entity and non-named entity tags are extracted to form the list *Stanford*.

WordNet The final heuristic makes use of WordNet (Fellbaum, 1998) which lists nouns that are often used as proper nouns with capitalisation. Nouns which appeared in both a capitalized and lowercased form were extracted to create the list *WordNet*.

Table 1 shows the number of nouns identified by each technique in the column labeled *words* which demonstrates that the number of nouns identified varies significantly depending upon which heuristic is used. A pairwise score is also shown to indicate the consistency between each list and two example lists, *BNCclaws* and *Gigaword*. It can be seen that the level of overlap is quite low and the various heuristics generate quite different lists of nouns. In particular the recall is low, in almost all cases less than a third of nouns in one list appear in the other.

One possible reason for the low overlap between the noun lists is mistakes by the heuristics used to extract them. For example, if a PoS tagger mistakenly tags just one instance of a common noun as proper then that noun will be added to the list extracted by the part of speech heuristic. Two filtering schemes were applied to improve the accuracy of the lists: (1) minimum frequency of occurrence, the noun must appear more than a set number of times in the corpus and (2) bias, the least common type of noun usage (i.e., common or proper) must account for more than a set percentage of all usages.

	words	BNCclaws		Gigaword	
		P	R	P	R
BNCclaws	41,110	100	100	31	2
BNCrasp	20,901	52	27	45	17
BNCcaps	18,524	56	26	66	21
5-grams	27,170	45	29	59	28
Gigaword	57,196	22	31	100	100
Wikipedia	7,351	49	9	59	8
WordNet	798	75	1	68	1
Stanford	64,875	43	67	26	29

Table 1: Pairwise comparison of lists. The nouns in each list are compared against the *BNCclaws* and *Gigaword* lists. Results are computed for P(recision) and R(ecall).

We experimented with various values for these filters and a selection of results is shown in Table 2, where *freq* is the minimum frequency of occurrence filter and *bias* indicates the percentage of the less frequent noun type.

	bias	freq	words	BNCclaws		Gigaword	
				P	R	P	R
BNCclaws	40	100	274	100	1	53	1
BNCrasp	30	100	253	94	1	85	0
5-grams	40	150	305	80	1	67	0
Stanford	40	200	260	87	1	47	0

Table 2: Pairwise comparison of lists with filtering

Precision (against *BNCclaws*) increased as the filters become more aggressive. However comparison with *Gigaword* does not show such high precision and recall is extremely low in all cases.

These experiments demonstrate that it is difficult to automatically generate a list of nouns that exhibit both common and proper usages. Manual analysis of the lists generated suggest that the heuristics can identify ambiguous nouns but intersecting the lists results in the loss of some obviously ambiguous nouns (however, their union introduces a large amount of noise). We select nouns from the lists created by these heuristics (such that the distribution of either the common or proper noun sense in the data was not less than 45%) for experiments in the following sections.[1]

[1]The 100 words selected for our evaluation are available at `http://pastehtml.com/view/cjsbs4xvl.txt`

3 Identifying Noun Types

We cast the problem of distinguishing between common and proper usages of nouns as a classification task and develop the following approaches.

3.1 Most frequent usage

A naive baseline is supplied by assigning each word its most frequent usage form (common or proper noun). The most frequent usage is derived from the training portion of labeled data.

3.2 n-gram system

A system based on n-grams was implemented using NLTK (Bird et al., 2009). Five-grams, four-grams, trigrams and bigrams from the training corpus are matched against a test corpus sentence, and results of each match are summed to yield a preferred use in the given context with a higher weight (experimentally determined) being assigned to longer n-grams. The system backs off to the most frequent usage (as derived from the training data).

3.3 Vector Space Model (VSM)

Distinguishing between common and proper nouns can be viewed as a classification problem. Treating the problem in this manner is reminiscent of techniques commonly employed in Word Sense Disambiguation (WSD). Our supervised approach is based on an existing WSD system (Agirre and Martinez, 2004) that uses a wide range of features:

- Word form, lemma or PoS bigrams and trigrams containing the target word.
- Preceding or following lemma (or word form) content word appearing in the same sentence as the target word.
- High-likelihood, salient, bigrams.
- Lemmas of all content words in the same sentence as the target word.
- Lemmas of all content words within a ± 4 word window of the target word.
- Non stopword lemmas which appear more than twice throughout the corpus.

Each occurrence of a common / proper noun is represented as a binary vector in which each position indicates the presence or absence of a feature. A centroid vector is created during the training phase for the common noun and the proper noun instances of a word. During the test phase, the centroids are compared to the vector of each test instance using the cosine metric, and the word is assigned the type of the closest centroid.

4 Evaluation

The approaches described in the previous section are evaluated on two data sets extracted automatically from the BNC. The **BNC-PoS** data set is created using the output from the CLAWS tagger. Nouns assigned the tag NP0 are treated as proper nouns and those assigned any other nominal tag as common nouns. (According to the BNC manual the NP0 tag has a precision 83.99% and recall 97.76%.[2]) This data set consists of all sentences in the BNC in which the target word appears. The second data set, **BNC-Capital**, is created using capitalisation information and consists of instances of the target noun that do not appear sentence-initially. Any instances that are capitalised are treated as proper nouns and those which are non-capitalised as common nouns.

Experiments were carried out using capitalised and decapitalized versions of the two test corpora. The decapitalised versions by lowercasing each corpus and using it for training and testing. Results are presented in Table 3. Ten fold cross validation is used for all experiments: i.e. 9/10th of the corpus were used to acquire the training data centroids and 1/10th was used for evaluation. The average performance over the 10 experiments is reported.

The vector space model (VSM) outperforms other approaches on both corpora. Performance is particularly high when capitalisation is included (*VSM w caps*). However, this approach still outperforms the baseline without case information (*VSM w/o caps*), demonstrating that using this simple approach is less effective than making use of local context.

[2]No manual annotation of common and proper nouns in this corpus exists and thus an exact accuracy figure for this corpus cannot be obtained.

	Gold standard	
	BNC-PoS	BNC-Capital
Most frequent	79%	67%
n-gram w caps	80%	77%
n-gram w/o caps	68%	56%
VSM w caps	90%	100%
VSM w/o caps	86%	80%

Table 3: BNC evaluation results

5 Applications

We also carried out experiments on two types of text in which capitalization information may not be available: social media and ASR output.

5.1 Twitter

As demonstrated in the BNC based evaluations, the system can be applied to text which does not contain capitalization information to identify proper nouns (and, as a side effect, enable the correction of capitalization). An example of such a dataset are the (up to) 140 character messages posted on Twitter.

There are some interesting observations to be made on messages downloaded from Twitter. Although some users choose to always tweet in lower case, the overall distribution of capitalization in tweets is high for the 100 words selected in Section 2 and only 3.7% of the downloaded tweets are entirely lower case. It also appeared that users who capitalize, do so fairly consistently.

This allows the creation of a dataset based on downloaded Twitter data[3]:

1. Identify purely lower case tweets containing the target word. These will form the test data (and are manually assigned usage).

2. Any non-sentence initial occurrences of the target word are used as training instances: lower case indicating a common instance, upper case indicating a proper instance.

14 words[4] were randomly selected from the list used in Section 4 and their lowercase tweet instances were manually annotated by a single annotator. The average proportion of proper nouns in the test data was 59%.

Training corpus	MF	n-grams	VSM
Twitter	59%	40%	60%
BNCclaw decap	59%	44%	79%

Table 4: Results on the Twitter data

The results for the three systems are presented in Table 4. As the length of the average sentence in the Twitter data is only 15 words (compared to 27 words in the BNCclaws data for the same target words), the Twitter data is likely to be suffering sparseness issues. This hypothesis is partly supported by the increase in performance when the BNCclaws decapitalized data is added to the training data, however, the performance of the n-gram system remains below the most frequent use. On closer examination, this is likely due to the skew in the data – there are many more examples for the common use of each noun, and thus each context is much more likely to have been seen in this setting.

5.2 Automatic speech recognition

Most automatic speech recognition (ASR) systems do not provide capitalization. However, our system does not rely on capitalization information, and therefore can identify proper / common nouns even if capitalization is absent. Also, once proper nouns are identified, the system can be used to restore case – a feature which allows an evaluation to take place on this dataset. We use the TDT2 Test and Speech corpus (Cieri et al., 1999), which contains ASR and a manually transcribed version of news texts from six different sources, to demonstrate the usefulness of this system for this task.

The ASR corpus is restricted to those segments which contain an equal number of target word occurrences in the ASR text and the manually transcribed version, and all such segments are extracted. The gold standard, and the most frequent usage, are drawn from the manually transcribed data.

Again, results are based on an average performance obtained using a ten fold cross validation. Three versions of training data are used: the 9/10 of ASR data (with labels provided by the manual transcription), the equivalent 9/10 of lowercased manu-

[3] `http://search.twitter.com/api`

[4] abbot, bull, cathedral, dawn, herald, justice, knight, lily, lodge, manor, park, president, raven and windows

Training corpus	MF	n-grams	VSM
Manual	66%	42%	73%
ASR	63%	41%	79%

Table 5: Results on the ASR data

ally transcribed data, and a combination of the two. The results can be seen in Table 5. The performance rise obtained with the VSM model when the ASR data is used is likely due to the repeated errors within this, which will not be appearing in the manually transcribed texts. The n-gram performance is greatly affected by the low volume of training data available, and again, a large skew within this.

6 Conclusion

We automatically generate lists of common and proper nouns using a number of different techniques. A vector space model technique for distinguishing common and proper nouns is found to achieve high performance when evaluated on the BNC. This greatly outperforms a simple n-gram based system, due to its better adaptability to sparse training data. Two application based evaluations also demonstrate the system's performance and as a side effect the system could serve as a technique for automatic case restoration.

Acknowledgments

The authors are grateful to the funding for this research received from Google (Google Research Award) and the UK Engineering and Physical Sciences Research Council (EP/J008427/1).

References

Agirre, E. and Martinez, D. (2004). The Basque Country University system: English and Basque tasks. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 44–48.

Baeza-Yates, R. and Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison Wesley Longman Limited, Essex.

Baldwin, T., Paul, M., and Joseph, A. (2009). Restoring punctuation and casing in English text. In *Proceedings of the 22nd Australian Joint Conference on Artificial Intelligence (AI09)*, pages 547–556.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit*. O'Reilly.

Brants, T. and Franz, A. (2006). Web 1T 5-gram v1.

Briscoe, T., Carroll, J., and Watson, R. (2006). The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*.

Chen, H., Huang, S., Ding, Y., and Tsai, S. (1998). Proper name translation in cross-language information retrieval. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 232–236, Montreal, Canada.

Cieri, C., Graff, D., Liberman, M., Martey, N., and Strassel, S. (1999). The TDT-2 text and speech corpus. In *Proceedings of DARPA Broadcast News Workshop*, pages 57–60.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database and some of its Applications*. MIT Press, Cambridge, MA.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, pages 363–370.

Garside, R. (1987). The CLAWS word-tagging system. In Garside, R., Leech, G., and Sampson, G., editors, *The Computational Analysis of English: A Corpus-based Approach*. London: Longman.

Graff, D. (2003). English Gigaword. Technical report, Linguistic Data Consortium.

Hermjakob, U., Knight, K., and Daumé III, H. (2008). Name translation in statistical machine translation - learning when to transliterate. In *Proceedings of ACL-08: HLT*, pages 389–397, Columbus, Ohio.

Leech, G. (1992). 100 million words of English: the British National Corpus. *Language Research*, 28(1):1–13.

Lopez, A. (2008). Statistical machine translation. *ACM Computing Surveys*, 40(3):1–49.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, Vol. 2(1-2):pp. 1–135.

Petrović, S., Osborne, M., and Lavrenko, V. (2010). Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189, Los Angeles, California.

Wilson, T., Wiebe, J., and Hoffman, P. (2009). Recognizing contextual polarity: an exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(5).