Exploring termhood using language models

Jody Foo
Linköping University
Linköping, Sweden
jody.foo@liu.se

Abstract

Term extraction metrics are mostly based on frequency counts. This can be a problem when trying to extract previously unseen multi-word terms. This paper explores whether smoothed language models can be used instead. Although a simplistic use of language models is examined in this paper, the results indicate that with more refinement, smoothed language models may be used instead of unsmoothed frequency-count based termhood metrics.

1 Background

Terminology work is the process of creating, harmonizing and standardizing term banks. The process involves the use of human terminologists and domain experts to a high degree, which can be costly for even small sized (e.g. 300 terms) term banks.

Automatic term extraction (ATE) or automatic term recognition (ATR) is a research area where methods researched that can to some degree automate the task of finding term candidates from document collections.

For the discussions in this paper we will be considering ATE used to facilitate terminology work done by terminologists. Looking from above, a workflow may be as follows.

1. Extract term candidates from corpus
2. Let domain expert process term candidates
3. Let terminologists create a term bank

This paper concerns step 1 which can be broken down into the following smaller steps.

a) Extract phrases
b) Asses termhood of phrases
c) Output term candidates

The following assumptions are used in this paper regarding the context of terminology.

- Term banks are used to reduce misunderstandings, eliminate ambiguity and raise the efficiency of communication between domain experts within the same domain, and to aid non-experts to understand domain specific texts.
- Terms represent Concepts.
- Definitions are attached to Concepts, not to terms.
- Terminologists are detectives that work together with domain experts to maintain a consistent terminology within the domain.

1.1 Term ranking concepts

Term ranking metrics can be categorized in several ways. One facet divides metrics into contrastive and non-contrastive measures. The contrastive model was introduced by Basili et al (2001) and explicitly argues that distributional differences between different document collections can be used to say something about extracted phrases.

The concept of termhood was introduced by Kageura and Umino (1996) and is defined as "*The degree to which a stable lexical unit is related to some domain-specific concepts*.". Unithood was also introduced by Kageura and Umino (1996) and is defined as "*the degree of strength or stability of syntagmatic combinations and collocations*". Both Wong and Liu (2009), and Zhang et al (2008) provide good overviews of termhood and unithood related metrics such as C-Value/NC-Value (Frantzi et al, 1998), Weirdness (Ahmad et al, 1999), Termextractor (Sclano and Velardi, 2007).

The ideal goal regarding termhood is to find a metric that correlates perfectly with the concept of termhood. Such a metric does however not yet exist and it is quite probable that constructing such a metric is a near impossible task for several reasons; one of them being that the properties of terms are difficult to capture. With

regard to actual work done by terminologists, a termhood metric is quite artificial. Also, it is important to keep in mind that a usable term ranking metric does not necessarily measure termhood – i.e. it may not be necessary to use a termhood metric to implement a useful term extraction application.

1.2 Support Vector Machines

In this paper, a *Support Vector Machine classifier* is used in an attempt to classify phrases into *term candidates* and *non-term candidates*.

The framework used is the e1071 package for R[1] (Dimitriadou et al 2009), which interfaces with libSVM, a Support Vector Machines implementation (Chang and Lin, 2001).

Support Vector Machines were introduced by Boser, et al (1992) and is a linear classifier that can use kernels to also classify non-linear data.

2 Questions

The existing research on term extraction is focused on term extraction as a once-off process using relatively large document collections. However, in reality, one may want to perform term extraction on smaller document sets containing new unseen documents from a previously processed domain. This may present a problem for frequency-count based metrics for two reasons

1 The document set may be too small for frequency based term metrics to be of use.

2 The first problem may be solved using a larger document collection is used to produce the metric values for extracted words/phrases from the smaller document collection. However, previously unseen multi-word terms cannot be assigned a score.

One way of solving problem 2, may be to use probability and perplexity scores from smoothed n-gram language models instead. The key point here is that a smoothed language model can produce a probability score for a multi-word term that uses a combination of words that has never been seen in previous document collections. Language Models have not been used in this way to the author's knowledge.

However, Patry and Langlais (2005) used language models of POS tags to improve phrase extraction beyond ordinary POS pattern extraction.

The work described in this paper is a preliminary study on using smoothed n-gram (word) language models to capture termhood.

3 Dataset

In this paper, two corpora are used 1) the British National Corpus (BNC) (BNC Consortium, 2000) and 2) English patent texts from the C04B IPC subclass (lime; magnesia; slag; cements; compositions thereof) as well as a set of domain expert validated terms from the subclass (note: the list of validated terms is not complete). See Table 1 for details of the used patent corpus.

C04B statistic	Value
Number of segments (sentences)	96,390
Number of tokens	2,395,177
Number of characters	1,2836,222
Validated terms	2,677

Table 1 C04B patent document corpus in numbers

3.1 Language models

Both the BNC corpus and C04B corpus were lemmatized using the commercial tagger Conexor Machinese Syntax[2]. The lemmatized corpora were then processed using SRI Language Modeling Toolkit, which produced one n-gram language model per corpus (two language models in total).

4 Phrase extraction and validation

The phrases from the dataset first extracted using IPhraxtor, a phrase extractor developed at Fodina Language Technology AB. A randomly sampled subset was then validated with regard to term candidates and non-term candidates.

4.1 Phrase extraction

Using IPhraxtor, noun phrases were extracted from the C04B corpus resulting in 101,191 extracted phrases. Among these phrases, 2,143 of the validated terms were found.

4.2 Term candidate validation

A sample was then extracted for manual term candidate markup. The sample was processed in Microsoft Excel where a non-domain-expert classified the phrases as either *term candidates* or *non-term candidates*. Note that the classification is between term candidates and non-term candidates; not between term and non-term. The reason is that the process we want to improve

[1] http://www.r-project.org/

[2] http://www.connexor.eu/technology/machinese/machinesesyntax/

outputs term candidates, not terms. Below are the guidelines used during the manual validation.

1 When validating the phrase as a term candidate the whole phrase must be considered, not just a part of the phrase. E.g. the phrase "*mold temperature*" may be considered a term candidate, but not "*measure mold temperature*"

2 Non-term candidates are

   a grammatically incomplete phrases, e.g. "involves passage", "improves compressive strength"

   b phrases that contain non words, misspelled words, or tokenization errors, e.g. "die(51a", "grains)"

   c phrases that are obviously general language such as idioms and general collocations, e.g. "*infinite length*", "*major role*"

   d phrases containing numbers

   e phrases starting with a verb

   f chemical formulas, e.g. H20 are not terms. Names of chemicals however, are, e.g. hydrogen oxide.

   g phrases starting with a "subjective" or referring adjective, e.g. *desired*, *intended*, *indicated*. Quantifying adjectives however, are fine, e.g. *poor*

Regarding guideline 2c, it is still a decision that depends on the validators experience and knowledge. Therefore, it is recommended that validators are domain experts in at least one field. For example the word "*accurate*" might be classified as a non-term candidate by a validator not familiar with the term "*accuracy*" in e.g. the domain of machine learning. Regarding guideline 2e; no phrases starting with a verb were intentionally extracted, but POS-tagger errors resulted in a few such phrases being included.

5 Contrastive features

The validated, extracted phrases were annotated with several features using the previously created language models. Each phrase was given a logarithmic probability value (logProb) and a perplexity value (ppl), first using the BNC language model, then the domain specific C04B language model. A probability ratio using the $logProb_{C04B}/logProb_{BNC}$ was also calculated and added. Finally, each phrase was annotated with the number of words in the feature. Each phrase also belonged to the class term candidate or non-term candidate. All values were normalized to the scale of 0-1. The features are summarized in Table 2.

Feature	Description
class	term candidate/non-term candidate
number of words	number of words in phrase
$logProb_{BNC}$	logarithmic probability of phrase in BNC language model
ppl_{BNC}	perplexity value of phrase in BNC language model
$logProb_{C04B}$	logarithmic probability of phrase in C04B language model
ppl_{C04B}	perplexity value of phrase in C04B language model
logProbRatio	the ratio between $logProb_{C04B}$ and $logProb_{BNC}$

Table 2 Features used for SVM classification

6 Looking for patterns

To understand the results of the SVM classification experiment, extracted phrases were ordered by class (term candidates first) and plotting their corresponding feature values in graphs. Figures 2-4, are examples of such graphs. In Figure 1, the precision of the ordered list is presented. This just shows how many term candidates and how many non-term candidates are in the list (# correct stops increasing where the non-term candidates begin). From Figures 2-4 it is clear that there does not seem to be any visible correlation between the language model output and the phrases classified as term candidates.

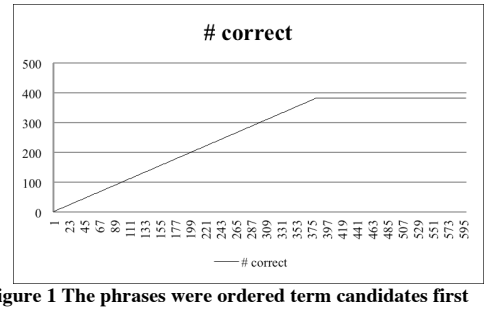

Figure 1 The phrases were ordered term candidates first

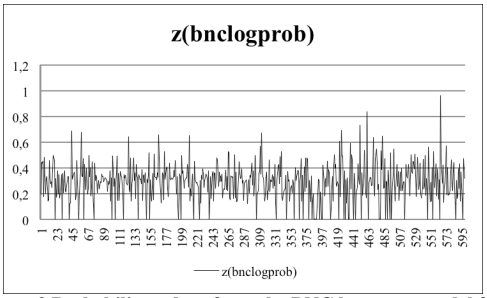

Figure 2 Probability values from the BNC language model for phrases ordered term candidates first

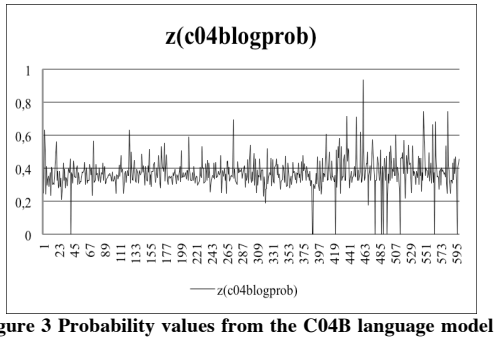

Figure 3 Probability values from the C04B language model for phrases ordered term candidates first

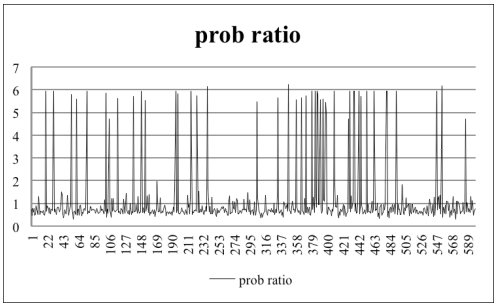

Figure 4 Probability ratio values for phrases ordered term candidates first

7 SVM classification results

A simple SVM experiment was conducted using the 1800 classified phrases. First a model was trained using 1200 of the phrases. Then the model was used to predict the class of the 600 phrases that were held back during training. The model used, predicted term candidates with a precision of 66.4% and a recall of 88.0%. Considering that the test partition contained 368 term candidate phrases, i.e. 61.3% of the test data were term candidates, the result of the classification is not much better than using the extracted phrases as they are.

8 Discussion and future work

Though the results from the classification experiment are not that strong, they were also the result of a rather simplistic use of language model provided features. The frequency count based metrics described in current research are still much more refined, as using the raw probability and perplexity values can be compared to using raw phrase frequency counts. Therefore, the author believes that there is more to gain from a language model approach. A higher level of refinement however is needed.

For example, a next step could be to consider phrases of different word length separately, as phrases containing more words have a lower probability in an n-gram language model by nature.

References

Ahmad, K., Gillam, L., & Tostevin, L. (1999). Weirdness Indexing for Logical Document Extrapolation and Retrieval. In *Proceedings of the Terminology and Artificial Intelligence Conference (TIA 2001)*.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *COLT '92 Proceedings of the fifth annual workshop on Computational learning theory* (p. 144-152).

Chang, C., & Lin, C. (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm (accessed 2011-04-25)

BNC Consortium. (2000). *The British National Corpus, version 2 (BNC World)* (2nd ed.). Oxford University Computing Services.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., & Weingessel, A. (2009). *Package "e1071."* Software available at http://cran.r-project.org/web/packages/e1071/index.html (accessed 2011-04-25)

Frantzi, K. T., Ananiadou, S., & Tsujii, J. (1998). The C-value/NC-value Method of Automatic Recognition for Multi-word Terms. In *ECDL '98* In *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries* (p. 585-604).

Kageura, K. (1996). Methods of Automatic Term Recognition. *Terminology*, *3*(2), 259-289(31).

Patry, A. & Langlais, P. (2005) Corpus-Based Terminology Extraction. In *Proceedings of the 7th International Conference on Terminology and Knowledge Engineering*, pp. 313-321

Sclano, F., & Velardi, P. (2007). TermExtractor: a Web Application to Learn the Common Terminology of Interest Groups and Research Communities. In *Proceedings of the 9th Conference on Terminology and Artificial Intelligence*.

Wong, W., & Liu, W. (2009). Determination of unithood and termhood for term recognition. In Song, M., & Brook Wu, Y. (Eds.), *Handbook of Research on Text and Web Mining Technologies. (pp. 500-529)*. IGI Global.

Zhang, Z., Iria, J., Brewster, C., & Ciravegna, F. (2008). A Comparative Evaluation of Term Recognition Algorithms. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*.