Stemming in the Language Modeling Framework

James Allan and Giridhar Kumaran
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003, USA
{allan,giridhar}@cs.umass.edu

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing— *Indexing methods*

General Terms

Experiments

Keywords

Stemming,Language Modeling

1. INTRODUCTION

Stemming is the process of collapsing words into their morphological root. For example, the terms *addicted*, *addicting*, *addictions*, *addictive*, and *addicts* might be conflated to their stem, *addict*. Over the years, numerous studies [2, 3, 4] have considered stemming as an external process — either to be ignored or used as a pre-processing step. In this study, we try and provide a fresh perspective to stemming. We are motivated by the observation that stemming can be viewed as a form of smoothing, as a way of improving statistical estimates. This suggests that stemming could be directly incorporated into a language model, which is what we achieve in this paper. Detailed discussions are available in[1].

2. STEMMING IN LANGUAGE MODELS

In this work, we focus on the query-likelihood variant of statistical language modeling [5]. Given a query $Q = q_1 q_2 q_3 \ldots q_n$, and a document $D = d_1 d_2 d_3 \ldots d_n$, the probability $P(Q|D)$ that the query would be generated by the document is $P(Q|D) = \prod_{j=1}^{n} P(q_j|D)$ with $P_{ML}(q_j|D) = \frac{c(q_j;D)}{\sum_{i=1}^{n} c(w_i;D)}$ where $c(q_i;D)$ represents the number of times that term q_i occurs in document D and ML refers to maximum likelihood. We use Jelinek Mercer smoothing. The normal language model without stemming is:

$$P_{\mathrm{u}}(w|D) = P_{\mathrm{ml}}(w|D) = \frac{c(w;D)}{\sum_{i=1}^{n} c(w_i;D)} \quad (1)$$

(smoothed with a background corpus according to λ). The simplest way to incorporate stemming into the language model would be to stem the collection before indexing—i.e., to index documents consisting of word stems. Short of doing that, we can simulate a stemmed collection at query-time by calculating the probability of a word using all words in its stem class. This leads to:

$$P_{\mathrm{s}}(w|D) = \frac{\sum_{w_j \in E(w)} c(w_j;D)}{\sum_{i=1}^{n} c(w_i;D)} = \sum_{w_i \in E(w)} P_{\mathrm{u}}(w_i|D) \quad (2)$$

(smoothed with a background corpus according to λ), where $E(w)$ represents the equivalence or stem class of w—that is, all words w_i that have the same stem as w (obviously, $w \in E(w)$).

2.1 Partial Stemming

Equations 1 and 2 are both ways of estimating the probability of occurrence of a word (though the latter actually estimates the probability of the word class), which immediately suggests the possibility of combining those estimates by interpolation:

$$P_{\mathrm{partial}}(w|D) = \alpha P_{\mathrm{u}}(w|D) + (1-\alpha) P_{\mathrm{s}}(w|D) \quad (3)$$

This combination allows a system to progress smoothly between stemming ($\alpha = 0$) and no stemming ($\alpha = 1$), and begs the question of what it means for α to lie in the middle of the range. While we found values of α that improved over plain stemming, we were unsuccessful in devising a strategy for estimating a value for α that would work over a range of collections.

2.2 Stemming as Smoothing

Rewriting Equation 2 we get

$$P(w|D) = P_{\mathrm{ml}}(w|D) + \sum_{\substack{w_i \in E(w) \\ w_i \neq w}} P_{\mathrm{ml}}(w_i|D)$$

The conversion to smoothing is accomplished by adding an interpolation parameter β:

$$P(w|D) = \beta P_{\mathrm{ml}}(w|D) + (1-\beta) \sum_{\substack{w_i \in E(w) \\ w_i \neq w}} P_{\mathrm{ml}}(w_i|D)$$

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.

For example, when $\beta = 1$, the equation reduces to unstemmed retrieval, and when $\beta = 0.5$ all words in the stem class (including the original word w) are treated equally. Thus the probability of a word is now calculated as an interpolation of its own probability and the probability of all other words in its stem class.

2.3 Ad-hoc Mixture Models

We generalize the view of stemming in the previous section as follows:

$$P_{\text{mix}}(w|D) = \frac{1}{\sum_j f(w_j, w)} \sum_{w_i \in E(w)} f(w_i, w) P_{\text{ml}}(w_i|D)$$

where $f(w_i, w)$ is a function that indicates how much significance term w_i has in calculating the probability for word w. This results in an interpolation over the probabilities for all of the words in the stem class. This general form admits a range of possibilities depending on how $f()$ is defined.

We consider two variants of P_{mix} that use different forms of $f()$. In the first, we use the co-occurrence analysis of corpus-based stemming to determine the value of $f()$. In the second, we imagine large numbers of stemmers and let $f()$ represent the chance that words would be put together by those stemmers.

2.4 Generative Models

As the first of two approaches, consider this process for generating a (query) word given a document model. First we generate a random word w_i from the vocabulary. Then we select a (query) word w (which might be w_i itself) that is a morphological variant of w_i and output w. This term generative model can be represented as:

$$P_{\text{tgen}}(w|D) = \sum_{w_i} P(w|w_i) P(w_i|D)$$

where $P(w_i|D)$ is the probability of selecting the word w_i from the model and $P(w|w_i)$ is the probability of selecting w as the morphological variant of w_i to output. We estimate the latter probability by the proportion of windows containing w_i that also contain w:

$$P(w|w_i) = n(w, w_i)/n(w_i)$$

This term generation approach is the same as the translation models that are common in language modeling approaches to cross-language retrieval [6]. The difference is that any "translations" are done within the same language and are restricted to words within the same stem class.

Another type of generative model is based on the intuition that a writer might think of a concept and then choose the appropriate variant of that concept depending on the situation. Specifically, the model first generates a stem class c and then selects from that class one of its words w to output—by earlier notation, $c = E(w)$, but we choose the class and then the word. Formally,

$$P_{\text{cgen}}(w|D) = \sum_{c} P(w|c) P(c|D)$$

where $P(c|D)$ represents the probability of choosing a particular class and $P(w|c)$ is the chance that the word w would be chosen. We estimate the latter as the collection frequency of the term w divided by the sum of the collection frequencies of all the terms in the equivalence class c. $P(c|D)$ can be calculated by counting the number of words in the stem class c that occur in D and dividing the the length of D.

Metric	Unstemmed	Stemmed	Ad-hoc Co-occur	Term Gen.
AvgPrec	0.2491	0.2725	0.2799	0.2779*
P@20	0.2389	0.2378	0.2441	0.2389

Table 1: Performance on the AP89 dataset of some discussed algorithms, measured in terms of average precision and precision at 20 documents retrieved. An asterisk indicates results that are significantly different ($P < 0.05$) from the stemmed results.

3. EXPERIMENTS

Our experiments with different stemming models show modest, but rarely statistically significant, improvements in comparison to the simplest form of stemming. All forms of stemming resulted in better accuracy than omitting stemming.

4. CONCLUSIONS

The work in this paper has not resulted in a vast improvement in stemming capabilities. However, we hope that by treating stemming in a range of probabilistic ways, some aspects may be better illuminated. For example, this approach suggests that we might be able to make use of better estimates of the probability of two words being in the same class. It also indicates that having more information about the probability of a particular morphological variant being chosen (which words are more common) can be readily incorporated.

5. ACKNOWLEDGMENTS

We wish to thank Jay Ponte for his initial observation that led to the research discussed in this paper.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

6. REFERENCES

[1] J. Allan and G. Kumaran. Details on stemming in the language modeling framework. In *UMass Amherst CIIR Tech. Report – IR-289*, 2003.

[2] D. Harman. How effective is suffixing? *J. American Society for Information Science*, 42(1):7–15, 1991.

[3] D. A. Hull. Stemming algorithms: A case study for detailed evaluation. *J. American Society for Information Science*, 47(1):70–84, 1996.

[4] L. Larkey, L. Ballesteros, and M. Connell. Improving stemming for arabic information retrieval: Light stemming and co-occurrence analysis. In *Proceedings of ACM SIGIR*, pages 269–274, 2002.

[5] J. Ponte and W. Croft. A language modeling approach to information retrieval. In *Proceedings of ACM SIGIR98*, pages 275–281, 1998.

[6] J. Xu, R. Weischdel, and C. Nguyen. Evaluating a probabilistic model for cross-lingual information retrieval. In *Proceedings of ACM SIGIR2001*, pages 105–109, 2001.