Abstract

We present a new approach to extracting keyphrases based on statistical language models. Our approach is to use pointwise KL-divergence between multiple language models for scoring both phraseness and informativeness, which can be unified into a single score to rank extracted phrases.

1 Introduction

In many real world deployments of text mining technologies, analysts are required to deal with large collections of documents from unfamiliar domains. Familiarity with the domain is necessary in order to get full leverage from text analysis tools. However, browsing data is not an efficient way to get an understanding of the topics and events which are particular to a domain.

For example, an analyst concerned with the area of hybrid cars may harvest messages from online forums. They may then want to rapidly construct a hierarchy of topics based on the content of these messages. In addition, in cases where these messages are harvested via a search of some sort, there is a requirement to obtain a rich and effective set of search terms.

Figure 1: Top 40 keyphrases automatically extracted from messages relevant to “civic hybrid” using our system

In order to capture domain-specific terms efficiently in limited time, the extraction result should be ranked with more indicative and good phrase first, as shown in this example.

2 Phraseness and informativeness

The word keyphrase implies two features: phraseness and informativeness.

Phraseness is a somewhat abstract notion which describes the degree to which a given word sequence is considered to be a phrase. In general, phraseness is defined by the user, who has his own criteria for the target application. For instance, one user might want only noun phrases while another user might be interested only in phrases describing a certain set of products. Although there is no single definition of the term phrase, in this paper, we focus on collocation or cohesion of consecutive words.

Informativeness refers to how well a phrase cap-
tures or illustrates the key ideas in a set of documents. Because informativeness is defined with respect to background information and new knowledge, users will have different perceptions of informativeness. In our calculations, we make use of the relationship between foreground and background corpora to formalize the notion of informativeness.

The target document set from which representative keyphrases are extracted is called the foreground corpus. The document set to which this target set is compared is called the background corpus. For example, a foreground corpus of the current week’s news would be compared to a background corpus of an entire news article archive to determine that certain phrases, like “press conference” are typical of news stories in general and do not capture the particulars of current events in the way that “national museum of antiquities” does.

Other examples of foreground and background corpora include: a web site for a certain company and web data in general; a newsgroup and the whole Usenet archive; and research papers of a certain conference and research papers in general.

In order to get a ranked keyphrase list, we need to combine both phraseness and informativeness into a single score. A sequence of words can be a good phrase but not an informative one, like the expression “in spite of.” A word sequence can be informative for a particular domain but not a phrase; “toyota, honda, ford” is an example of a non-phrase sequence of informative words in a hybrid car domain. The algorithm we propose for keyphrase finding requires that the keyphrase score well for both phraseness and informativeness.

3 Related work

Word collocation Various collocation metrics have been proposed, including mean and variance (Smadja, 1994), the t-test (Church et al., 1991), the chi-square test, pointwise mutual information (MI) (Church and Hanks, 1990), and binomial log-likelihood ratio test (BLRT) (Dunning, 1993).

According to (Manning and Schütze, 1999), BLRT is one of the most stable methods for collocation discovery. (Pantel and Lin, 2001) reports, however, that BLRT score can be also high for two frequent terms that are rarely adjacent, such as the word pair “the the,” and uses a hybrid of MI and BLRT.

Keyphrase extraction Damerau (1993) uses the relative frequency ratio between two corpora to extract domain-specific keyphrases. One problem of using relative frequency is that it tends to assign too high a score for words whose frequency in the background corpus is small (or even zero).

Some work has been done in extracting keyphrases from technical documents treating keyphrase extraction as a supervised learning problem (Frank et al., 1999; Turney, 2000). The portability of a learned classifier across various unstructured/structured text is not clear, however, and the agreement between classifier and human judges is not high.1

We would like to have the ability to extract keyphrases from a totally new domain of text without building a training corpus.

Combining keyphrase and collocation Yamamoto and Church (2001) compare two metrics, MI and Residual IDF (RIDF), and observed that MI is suitable for finding collocation and RIDF is suitable for finding informative phrases. They took the intersection of each top 10% of phrases identified by MI and RIDF, but did not extend the approach to combining the two metrics into a unified score.

4 Baseline method based on binomial log-likelihood ratio test

We can use various statistics as a measure for phraseness and informativeness. For our baseline, we have selected the method based on binomial log-likelihood ratio test (BLRT) described in (Dunning, 1993).

The basic idea of using BLRT for text analysis is to consider a word sequence as a repeated sequence of binary trials comparing each word in a corpus to a target word, and use the likelihood ratio of two hypotheses that (i) two events, observed \( k_1 \) times out of \( n_1 \) total tokens and \( k_2 \) times out of \( n_2 \) total tokens respectively, are drawn from different distributions and (ii) from the same distribution.

1 e.g. Turney reports 62% “good”, 18% “bad”, 20% “no opinion” from human judges.
The BLRT score is calculated with

$$2 \log \frac{L(p_1, k_1, n_1)L(p_2, k_2, n_2)}{L(p, k_1, n_1)L(p, k_2, n_2)}$$

where \( p_i = k_i/n_i, p = (k_1 + k_2)/(n_1 + n_2) \), and

$$L(p, k, n) = p^k(1-p)^{n-k}$$

In the case of calculating the phraseness score of an adjacent word pair \((x, y)\), the null hypothesis is that \(x\) and \(y\) are independent, which can be expressed as \(p(y|x) = p(y|\neg x)\). We can use Equation (1) to calculate phraseness by setting:

$$k_1 = C(x, y),$$
$$n_1 = C(x),$$
$$k_2 = C(\neg x, y) = C(y) - C(x, y),$$
$$n_2 = C(\neg x) = \sum_w C(w) - C(x)$$

where \(C(x)\) is the frequency of the word \(x\) and \(C(x, y)\) is the frequency of \(y\) following \(x\).

For calculating informativeness of a word \(w\),

$$k_1 = C_{fg}(w),$$
$$n_1 = \sum_w C_{fg}(w),$$
$$k_2 = C_{bg}(w),$$
$$n_2 = \sum_w C_{bg}(w)$$

where \(C_{fg}(w)\) and \(C_{bg}(w)\) are the frequency of \(w\) in the foreground and background corpus, respectively.

Combining a phraseness score \(\varphi_p\) and an informativeness score \(\varphi_i\) into a single score value is not a trivial task since the BLRT scores vary a lot between phraseness and informativeness and also depending on data (c.f. Figure 6 (a)).

One way to combine those scores is to use an exponential model. We experimented with the following logistic function:

$$\varphi = \frac{1}{1 + \exp(-a\varphi_p - b\varphi_i + c)}$$

whose parameters \(a, b,\) and \(c\) are estimated on a held-out data set, given feedback from users (i.e. supervised).

Figure 2 shows some example phrases extracted with this method from the data set described in Section 6.1, where the parameters, \(a, b, c\), are manually optimized on the test data.

Although it is possible to rank keyphrases using this approach, there are a couple of drawbacks.

### Necessity of tuning parameters
The existence of parameters in the combining function requires human labeling, which is sometimes an expensive task to do, and the robustness of learned weight across domains is unknown. We would like to have a parameter-free and robust way of combining scores.

### Inappropriate symmetry
BLRT tests to see if two random variables are independent or not. This sometimes leads to unwanted phrases getting a high score. For example, when the background corpus happens to have many occurrences of phrase \(al\ jazeera\) which is an unusual phrase in the foreground corpus, then the phrase still gets high score of informativeness because the distribution is so different. What we would like to have instead is asymmetric scoring function to test the loss of the action of not taking the target phrase as a keyphrase.

In the next section, we propose a new method trying to address these issues.

### 5 Proposed method

#### 5.1 Language models and expected loss

A language model assigns a probability value to every sequence of words \(w = w_1w_2\ldots w_n\). The probability \(P(w)\) can be decomposed as

$$P(w) = \prod_{i=1}^{n} P(w_i|w_1w_2\ldots w_{i-1})$$

Assuming \(w_i\) only depends on the previous \(N\) words, \(N\)-gram language models are commonly
used. The following is the trigram language model case.

\[ P(w) = \prod_{i=1}^{n} P(w_i|w_{i-2}, w_{i-1}) \]

Here each word only depends on the previous two words. Please refer to (Jelinek, 1990) and (Chen and Goodman, 1996) for more about N-gram models and associated smoothing methods.

Now suppose we have a foreground corpus and a background corpus and have created a language model for each corpus. The simplest language model is a unigram model, which assumes each word of a given word sequence is drawn independently. We denote the unigram model for the foreground corpus as \( L^{1}_{fg} \) and for the background corpus as \( L^{1}_{bg} \). We can also train higher order models \( L^{N}_{fg} \) and \( L^{N}_{bg} \) for each corpus, each of which is a \( N \)-gram model, where \( N(>1) \) is the order.

\[ \delta_{\text{w}}(p \parallel q) \text{ def } = p(w) \log \frac{p(w)}{q(w)} \]  

Intuitively, this is the contribution of the phrase \( w \) to the expected loss of the entire distribution.

We can now quantify phraseness and informativeness as follows:

**Phraseness** of \( w \) is how much we lose information by assuming independence of each word by applying the unigram model, instead of the \( N \)-gram model.

\[ \delta_{\text{w}}(L^{N}_{fg} \parallel L^{1}_{fg}) \]  

**Informativeness** of \( w \) is how much we lose information by assuming the phrase is drawn from the background model instead of the foreground model.

\[ \delta_{\text{w}}(L^{N}_{fg} \parallel L^{N}_{bg}), \text{ or } \delta_{\text{w}}(L^{1}_{fg} \parallel L^{1}_{bg}) \]  

**Combined** The following is considered to be a mixture of phraseness and informativeness.

\[ \delta_{\text{w}}(L^{N}_{fg} \parallel L^{1}_{bg}) \]
Note that the KL divergence is always non-negative, but the pointwise KL divergence can be a negative value. An example is the phraseness of the bigram “the the”.

\[ p(\text{the, the}) \log \frac{p(\text{the, the})}{p(\text{the})p(\text{the})} < 0 \]

since \( p(\text{the, the}) < p(\text{the})p(\text{the}) \).

Also note that in the case of phraseness of a bigram, the equation looks similar to pointwise mutual information (Church and Hanks, 1990), but they are different. Their relationship is as follows.

\[ \delta_{w}(p(x, y) || p(x)p(y)) = p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \]

The pointwise KL divergence does not assign a high score to a rare phrase, whose contribution of loss is small by definition, unlike pointwise mutual information, which is known to have problems (as described in (Manning and Schütze, 1999), e.g.).

5.3 Combining phraseness and informativeness

One way of getting a unified score of phraseness and informativeness is using equation (11). We can also calculate phraseness and informativeness separately and then combine them.

We combine the phraseness score \( \varphi_p \) and informativeness score \( \varphi_i \) by simply adding them into a single score \( \varphi \).

\[ \varphi = \varphi_p + \varphi_i \]  

(12)

Intuitively, this can be thought of as the total loss. We will show some empirical results to justify this scoring in the next section.

6 Experimental results

In this section, we show some preliminary experimental results of applying our method on real data.

6.1 Data set

We used the 20 newsgroups data set, which contains 20,000 messages (7.4 million words) between February and June 1993 taken from 20

USenet newsgroups, as the background data set, and another 20,000 messages (4 million words) between June and September 2002 taken from rec.arts.movies.current-films newsgroup as the foreground data set. Each message’s subject header and the body of the message (including quoted text) is tokenized into lowercase tokens on both data set. No stemming is applied.

6.2 Finding key-bigrams

The first experiment we show is to find key-bigrams, which is the simplest case requiring combination of phraseness and informativeness scores. Figure 4 outlines the extraction procedure.

- Inputs: foreground and background corpus.
- 1. create background language model from the background corpus.
- 2. count all adjacent word pairs in the foreground corpus, skipping pre-annotated boundaries (such as HTML tag boundaries) and stopwords.
- 3. for each pair of words \((x, y)\) in the count, calculate phraseness from \( p(x, y)_f \) and \( p(x)_f p(y)_f \) and informativeness from \( p(x, y)_b \) and \( p(x)_b p(y)_b \). Add the two score values as the unified score.
- 4. sort the results by the unified score.

Output: a list of key-bigrams ranked by unified score.

Figure 4: Procedure to find key-bigrams

For this experiment we used unsmoothed count for calculating phraseness \( p(x, y) = C(x, y)/N, p(w) = C(w)/N \) where \( N = \sum_x C(x) = \sum_{x, y} C(x, y) \), and used the unigram model for calculating informativeness with Katz smoothing (Chen and Goodman, 1996) to handle zero occurrences.

Figure 5 shows the extracted key-bigrams using this method. Comparing to Figure 2, you can see that those two methods extract almost identical ranked phrases. Note that we needed to tune three parameters to combine phraseness and informativeness in BLRT, but no parameter tuning was required in this method.

The reason why “message news” becomes the top phrase in both methods is that it appears frequently enough in message citation headers such...
One question that might be asked is “what if we just sort by frequency?”. If we sort by frequency, “blair witch project” is 92nd and “empire strikes back” is 110th on the ranked list. Since the longer the phrase becomes, the lower the frequency of the phrase is, it is not an appropriate method for ranking phrases.

Figure 7: Result of re-ranking output from the phrase extension module

6.4 Revisiting unigram informativeness

An alternative approach to calculate informativeness from the foreground LM and the background LM is just to take the ratio of likelihood scores, \( \frac{p_f(w)}{p_b(w)} \). This is a smoothed version of relative frequency ratio which is commonly used to find subject-specific terms (Damerau, 1993).

Figure 8 compares extracted keywords ranked with pointwise KL and likelihood ratio scores, both of which use the same foreground and background unigram language model. We used messages retrieved from the query *Infiniti G35* as the foreground corpus and the same 20 newsgroup data as the background corpus. Katz smoothing is applied to both language models.

As we can see, those two methods return very different ranked lists. We think the pointwise KL returns a set of keywords closer to human judgment.

One example is the word “infiniti”, which we expected to be one of the informative words since it is the query word. The pointwise KL score picked the word as the third informative word, but the likelihood score missed it. Whereas “6mt”, picked up by the likelihood ratio, which occurs 37 times in the
foreground corpus and none in the background corpus does not seem to be a good keyword.

The following table shows statistics of those two words:

| token | \( p_{\text{fg}}(w) \) | \( p_{\text{bg}}(w) \) | PKL | LR |
|-------|-----------------|-----------------|-----|-----|
| 6mt   | 1.837E-4        | 8.705E-8        | 0.0020 | 2110 |
| infiniti | 2.269E-3      | 4.475E-6        | 0.0204 | 506 |

Since the likelihood of “6mt” with respect to the background LM is so small, the likelihood ratio of the word becomes very large. But the pointwise KL score discounts the score appropriately by considering that the frequency of the word is low. Likelihood ratio (or relative frequency ratio) has a tendency to pick up rare words as informative. Pointwise KL seems more robust in sparse data situations.

One disadvantage of the pointwise KL statistic might be that it also picks up stopwords or punctuation, when there is a significant difference in style of writing, etc., since these words have significantly high frequency. But stopwords are easy to define or can be generated automatically from corpora, and we don’t consider this to be a significant drawback. We also expect a better background model and better smoothing mechanism could reduce the necessity of the stopword list.

## 7 Discussion

### Necessity of both phraseness and informativeness

Although phraseness itself is domain-dependent to some extent (Smadja, 1994), we have shown that there is little correlation between informativeness and phraseness scores.

### Combining method

One way to calculate a combined score is directly comparing \( LM_{\text{FG}}^N \) and \( LM_{\text{BG}}^1 \) in Figure 3. We have tried both approaches and got a better result from combining separate phraseness and informativeness scores. We think this is due to data sparseness of the higher order ngram in the background corpus. Further investigation is required to make a conclusion.

We have used the simplest method of combining two scores by adding them. We have also tried har-
monic mean and geometric mean but they did not improve the result. We could also apply linear interpolation to put more weight on one score value, or use an exponential model to combine score, but this will require tuning parameters.

Benefits of using a language model One benefit of using a language model approach is that one can take advantage of various smoothing techniques. For example, by interpolating with a character-based n-gram model, we can make the LM more robust with respect to spelling errors and variations. Consider the following variations, which we need to treat as a single entity: al-Qaida, al Qaida, al Qaeda, al Queda, al-Qa’ida, al Qa’ida (found in online sources). Since these are such unique spellings in English, character n-gram is expected to be able to give enough likelihood score to different spellings as well.

It is also easy to incorporate other models such as topic or discourse model, use a cache LM to capture local context, and a class-based LM for the shared concept. It is also possible to add a phrase length prior probability in the model for better likelihood estimation.

Another useful smoothing technique is linear interpolation of the foreground and background language models, when the foreground and background corpus are disjoint.

8 Conclusion

We have explained that phraseness and informativeness should be unified into a single score to return useful ranked keyphrases for analysts. Our proposed approach calculates both scores based on language models and unified into a single score. The phrases generated by this method are intuitively very useful, but the results are difficult to evaluate quantitatively.

In future work we would like to further explore evaluation of keyphrases, as well as investigate different smoothing techniques. Further extensions include developing a phrase boundary segmentation algorithm based on this framework and exploring applicability to other languages.

References

Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, Proc. 20th Int. Conf. Very Large Data Bases, VLDB, pages 487–499. Morgan Kaufmann, 12–15.

Stanley F. Chen and Joshua T. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In Proceedings of the 34th Annual Meeting of the ACL, pages 310–318, Santa Cruz, California, June.

Kenneth W. Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. In Computational Linguistics, volume 16.

K. Church, P. Hanks, D. Hindle, and W. Gale, 1991. Using Statistics in Lexical Analysis, pages 115–164. Lawrence Erlbaum.

Thomas M. Cover and Joy A. Thomas. 1991. Elements of Information Theory. John Wiley.

Fred J. Damerau. 1993. Generating and evaluating domain-oriented multi-word terms from texts. Information Processing and Management, 29(4):433–447.

Ted E. Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. Computational Linguistics, 19(1):61–74.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In IJCAI, pages 668–673.

Frederick Jelinek. 1990. Self-organized language modeling for speech recognition. In Alex Waibel and Kai-Fu Lee, editors, Readings in Speech Recognition, pages 450–506. Morgan Kaufmann Publishers, Inc., San Mateo, California.

Christopher D. Manning and Hinrich Schütze. 1999. Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, Massachusetts.

Patrick Pantel and Dekang Lin. 2001. A statistical corpus-based term extractor. In E. Stroulia and S. Matwin, editors, Lecture Notes in Artificial Intelligence, pages 36–46. Springer-Verlag.

Frank Z. Smadja. 1994. Retrieving collocations from text: Xtract. Computational Linguistics, 19(1):143–177.

Peter D. Turney. 2000. Learning algorithms for keyphrase extraction. Information Retrieval, 2(4):303–336.

Mikio Yamamoto and Kenneth W. Church. 2001. Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. Computational Linguistics, 27(1):1–30.