A Language Model Approach to Keyphrase Extraction

Takashi Tomokiyo and Matthew Hurst
Applied Research Center
Intelliseek, Inc.
Pittsburgh, PA 15213
{ttomokiyo,mhurst}@intelliseek.com

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.

The technology described in this paper is an example of a phrase finder capable of delivering a set of indicative phrases given a particular set of documents from a target domain.

In the hybrid car example, the result of this process is a set of phrases like that shown in Figure 1.

1	civic hybrid	21	mustang gt
2	honda civic hybrid	22	ford escape
3	toyota prius	23	steering wheel
4	electric motor	24	toyota prius today
5	honda civic	25	electric motors
6	fuel cell	26	gasoline engine
7	hybrid cars	27	internal combustion engine
8	honda insight	28	gas engine
9	battery pack	29	front wheels
10	sports car	30	key sense wire
11	civic si	31	civic type r
12	hybrid car	32	test drive
13	civic lx	33	street race
14	focus fcv	34	united states
15	fuel cells	35	hybrid powertrain
16	hybrid vehicles	36	rear bumper
17	tour de sol	37	ford focus
18	years ago	38	detroit auto show
19	daily driver	39	parking lot
20	jetta tdi	40	rear wheels

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 captures 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

$$2 \log \frac{L(p_1, k_1, n_1) L(p_2, k_2, n_2)}{L(p, k_1, n_1) L(p, k_2, n_2)} \tag{1}$$

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

$$L(p, k, n) = p^k (1 - p)^{n-k} \tag{2}$$

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:

$$\begin{aligned} k_1 &= C(x, y), \\ n_1 &= C(x), \\ k_2 &= C(\neg x, y) = C(y) - C(x, y), \\ n_2 &= C(\neg x) = \textstyle\sum_w C(w) - C(x) \end{aligned} \tag{3}$$

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,

$$\begin{aligned} k_1 &= C_{fg}(w), \\ n_1 &= \textstyle\sum_w C_{fg}(w), \\ k_2 &= C_{bg}(w), \\ n_2 &= \textstyle\sum_w C_{bg}(w) \end{aligned} \tag{4}$$

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

Combining a phraseness score φ_p and an informativeness score φ_i into a single score value is not a trivial task since the 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:

$$\varphi = \frac{1}{1 + \exp(-a\varphi_p - b\varphi_i + c)} \tag{5}$$

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.

1	message news	16	sixth sense
2	minority report	17	hey kids
3	star wars	18	gaza man
4	john harkness	19	lee harrison
5	derek janssen	20	years ago
6	robert frenchu	21	julia roberts
7	sean o'hara	22	national guard
8	box office	23	bourne identity
9	dawn taylor	24	metrotoday www.zap2it.com
10	anthony gaza	25	starweek magazine
11	star trek	26	eric chomko
12	ancient race	27	wilner starweek
13	scooby doo	28	tim gueguen
14	austin powers	29	jodie foster
15	home.attbi.com hey	30	johnnie kendricks

Figure 2: Keyphrases extracted with BLRT (a=0.0003, b=0.000005, c=8)

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 $\mathbf{w} = w_1 w_2 \ldots w_n$. The probability $P(\mathbf{w})$ can be decomposed as

$$P(\mathbf{w}) = \prod_{i=1}^{n} P(w_i | w_1 w_2 \ldots w_{i-1})$$

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.

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

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 LM^1_{fg} and for the background corpus as LM^1_{bg}. We can also train higher order models LM^N_{fg} and LM^N_{bg} for each corpus, each of which is a N-gram model, where $N(> 1)$ is the order.

$$\begin{tikzcd}
 & \longleftarrow \textit{informativeness} \longrightarrow & \\
\uparrow & LM^N_{\text{fg}} \arrow[r, dash] \arrow[d, dash] & LM^N_{\text{bg}} \arrow[d, dash] \\
\textit{phraseness} & & \\
\downarrow & LM^1_{\text{fg}} \arrow[r, dash] & LM^1_{\text{bg}}
\end{tikzcd}$$

Figure 3: Phraseness and informativeness as loss between language models.

Among those four models, LM^N_{fg} will be the best model to describe the foreground corpus in the sense that it has the smallest cross-entropy or perplexity value over the corpus.

If we use one of the other three models instead, then we have some inefficiency or loss to describe the corpus. We expect the amount of loss between using LM^N_{fg} and LM^1_{fg} is related to phraseness and the loss between LM^N_{fg} and LM^N_{bg} is related to informativeness. Figure 3 illustrates these relationships.

5.2 Pointwise KL-divergence between models

One natural metric to measure the loss between two language models is the Kullback-Leibler (KL) divergence. The KL divergence (also called relative entropy) between two probability mass function $p(x)$ and $q(x)$ is defined as

$$D(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{6}$$

KL divergence is "*a measure of the inefficiency of assuming that the distribution is q when the true distribution is p.*" (Cover and Thomas, 1991)

You can see this by the following relationship:

$$\begin{aligned} D(p \parallel q) &= \sum_x p(x) \log p(x) - \sum_x p(x) \log q(x) \\ &= \sum_x p(x) \frac{1}{\log q(x)} - H(X) \end{aligned}$$

The first term $\sum_x p(x) \frac{1}{\log q(x)}$ is the cross entropy and the second term $H(X)$ is the entropy of the random variable X, which is how much we could compress symbols if we know the true distribution p.

We define *pointwise KL divergence* $\delta_{\mathbf{w}}(p \parallel q)$ to be the term inside of the summation of Equation (6):

$$\delta_{\mathbf{w}}(p \parallel q) \stackrel{\text{def}}{=} p(\mathbf{w}) \log \frac{p(\mathbf{w})}{q(\mathbf{w})} \tag{7}$$

Intuitively, this is the contribution of the phrase $\mathbf{w}$ to the expected loss of the entire distribution.

We can now quantify phraseness and informativeness as follows:

Phraseness of $\mathbf{w}$ is how much we lose information by assuming independence of each word by applying the unigram model, instead of the N-gram model.

$$\delta_{\mathbf{w}}(LM^N_{\text{fg}} \parallel LM^1_{\text{fg}}) \tag{8}$$

Informativeness of $\mathbf{w}$ is how much we lose information by assuming the phrase is drawn from the background model instead of the foreground model.

$$\delta_{\mathbf{w}}(LM^N_{\text{fg}} \parallel LM^N_{\text{bg}}), \text{ or} \tag{9}$$

$$\delta_{\mathbf{w}}(LM^1_{\text{fg}} \parallel LM^1_{\text{bg}}) \tag{10}$$

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

$$\delta_{\mathbf{w}}(LM^N_{\text{fg}} \parallel LM^1_{\text{bg}}) \tag{11}$$

Note that the KL divergence is always non-negative[2], but the pointwise KL divergence can be a negative value. An example is the phraseness of the bigram "*the the*".

$$p(\mathit{the}, \mathit{the}) \log \frac{p(\mathit{the}, \mathit{the})}{p(\mathit{the})p(\mathit{the})} < 0$$

since $p(\mathit{the}, \mathit{the}) \ll p(\mathit{the})p(\mathit{the})$.

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.

$$\delta_{\mathbf{w}}(p(x,y) \parallel p(x)p(y)) = p(x,y) \underbrace{\log \frac{p(x,y)}{p(x)p(y)}}_{\text{pointwise MI}}$$

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 φ_p and informativeness score φ_i by simply adding them into a single score φ.

$$\varphi = \varphi_p + \varphi_i \tag{12}$$

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[3], 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 $p(x,y)_{\text{fg}}$ and $p(x)_{\text{fg}}p(y)_{\text{fg}}$ and informativeness from $p(x,y)_{\text{fg}}$ and $p(x,y)_{\text{bg}}$. 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)[4] 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

[2] from Jensen's inequality.

[3] `http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html`

[4] with cutoff $K = 5$

1	message news	16	hey kids
2	minority report	17	years ago
3	star wars	18	gaza man
4	john harkness	19	sixth sense
5	robert frenchu	20	lee harrison
6	derek janssen	21	julia roberts
7	box office	22	national guard
8	sean o'hara	23	bourne identity
9	dawn taylor	24	metrotoday www.zap2it.com
10	anthony gaza	25	starweek magazine
11	star trek	26	eric chomko
12	ancient race	27	wilner starweek
13	home.attbi.com hey	28	tim gueguen
14	scooby doo	29	jodie foster
15	austin powers	30	kevin filmnutboy

Figure 5: Key-bigrams extracted with pointwise KL

as `John Smith ⟨js@foo.com⟩ wrote in message news:1pk0a@foo.com`, which was not common in the 20 newsgroup dataset.[5] A more sophisticated document analysis tool to remove citation headers is required to improve the quality further.

Figure 6 shows the distribution of phraseness and informativeness scores of bigrams extracted using the BLRT and pointwise KL methods. One can see that there is little correlation between phraseness and informativeness in both ranking methods. Also note that the range of x and y axis is very different in BLRT, but in the pointwise KL method they are comparable ranges. That makes combining two scores easy in the pointwise KL approach.

6.3 Ranking n-length phrases

The next example is ranking n-length phrases. We applied a phrase extension algorithm based on the APriori algorithm (Agrawal and Srikant, 1994) to the output of the key-bigram finder in the previous example to generate n-length candidates whose frequency is greater than 5, then applied a linguistic filter which rejects phrases that do not occur in valid noun-phrase contexts (e.g. following articles or possessives) at least once in the corpus. We ranked resulting phrases using pointwise KL score, using the same smoothing method as in the bigram case.

Figure 7 shows the result of re-ranking keyphrases extracted from the same movie corpus. We can see that bigrams and trigrams are interleaved in natural order (although not many long phrases are extracted from the dataset, since longer NP did not occur more than five times). Figure 1 was another example of the result of the same pipeline of methods.

[5] a popular citation pattern in 1993 was "In article ⟨1pk0a@foo.com⟩, js@foo.com (John Smith) writes:"

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, frequency is not an appropriate method for ranking phrases.

1	minority report	21	bad guy
2	box office	22	country bears
3	scooby doo	23	man's man
4	sixth sense	24	long time
5	national guard	25	spoiler space
6	bourne identity	26	empire strikes back
7	air national guard	27	top ten
8	united states	28	politically correct
9	phantom menace	29	white people
10	special effects	30	tv show
11	hotel room	31	bad guys
12	comic book	32	freddie prinze jr
13	blair witch project	33	monster's ball
14	short story	34	good thing
15	real life	35	evil minions
16	jude law	36	big screen
17	iron giant	37	political correctness
18	bin laden	38	martial arts
19	black people	39	supreme court
20	opening weekend	40	beautiful mind

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

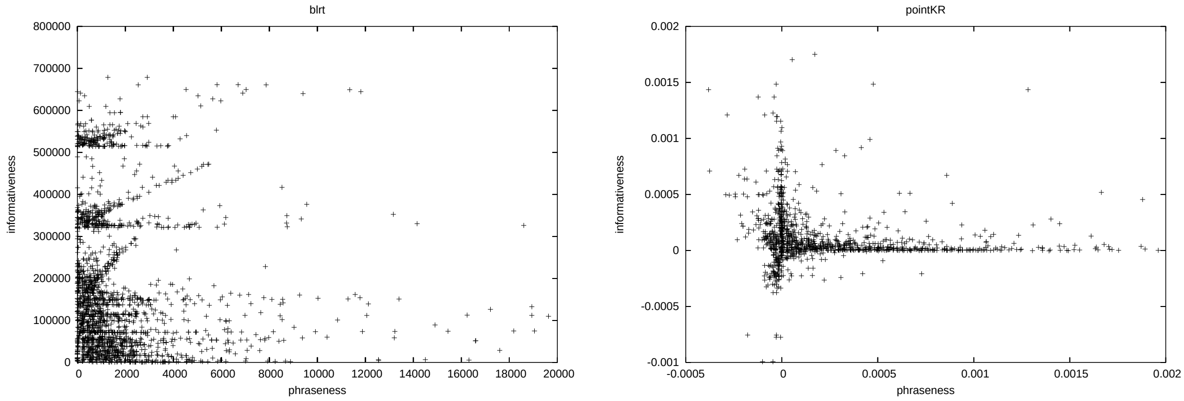
6.4 Revisiting unigram informativeness

An alternative approach to calculate informativeness from the foreground LM and the background LM is just to take the ratio of likelihood scores, $p_{\text{fg}}(\mathbf{w})/p_{\text{bg}}(\mathbf{w})$. 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



(a) BLRT (b) LM-pointKL

Figure 6: Phraseness and informativeness score of bigrams extracted with BLRT (a) and pointwise KL divergence between LMs (b).

	point KL		likelihood ratio	
rank	freq	term	freq	term
1	1599	g35	1599	g35
2	1145	car	156	330i
3	450	infiniti	117	350z
4	299	coupe	113	doo
5	299	nissan	90	wrx
6	383	bmw	76	is300
7	156	330i	47	willow
8	441	cars	39	rsx
9	248	sedan	37	6mt
10	331	originally	35	scooby
11	201	altima	35	s2000
12	117	350z	33	gt-r
13	113	doo	32	lol
14	235	sport	30	heatwave
15	172	maxima	28	g22
16	90	wrx	26	gtr
17	111	skyline	23	g21
18	76	is300	23	g17
19	186	honda	23	nsx
20	221	engine	22	tl-s

Figure 8: Top 20 keywords extracted using pointwise-KL and likelihood ratio (after stopwords removed) from messages retrieved from the query "*Infiniti G35*"

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:[6]

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.

[6]"infiniti" occurs 34 times in the "rec.autos" section of the 20 newsgroup data set.

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^N_{fg} and LM^1_{bg} 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 harmonic 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-Qaeda, 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 Maeio, 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.