A comparison of summarisation methods based on term specificity estimation

Constantin Orăsan, Viktor Pekar and Laura Hasler

Research Group in Computational Linguistics
School of Humanities, Languages and Social Sciences
University of Wolverhampton
Stafford St., Wolverhampton, WV1 1SB
{C.Orasan, V.Pekar, L.Hasler}@wlv.ac.uk

Abstract

In automatic summarisation, knowledge poor methods do not necessarily perform worse than those which employ several knowledge sources to produce a summary. This paper presents a comprehensive comparison of several summarisation methods based on term specificity estimation in order to find out which one performs best. Parameters such as quality of the summary produced and the resources required to produce accurate results are considered in order to find out which of these methods is more appropriate for a real world application. Intrinsic and extrinsic evaluation indicates that TF*RIDF, a variant of the commonly used TF*IDF, is the best performing method.

1. Introduction

In automatic summarisation it is quite often the case that methods which rely on a large number of resources do not necessarily perform better than those which use much fewer resources such as baselines (Brandow et al., 1995; Alonso i Alemany and Fuentes Fort, 2003). Moreover, as a result of the large number of modules which are combined to produce the summary, these more advanced methods are rather slow. At the opposite end of the spectrum are those methods where only a few modules are needed to be run in order to produce a summary and therefore they run much faster. Because of this, such methods could be very suitable for real-world applications, provided that they produce summaries of an acceptable quality.

In this paper, we compare several methods which rely on term specificity estimation in order to identify which is the best one. In addition, we also investigate the kind of resources these methods require in order to produce accurate results. It should be emphasised that this paper does not try to suggest that the development of more linguistically justified methods is not useful. Our aim here is to identify which of the existing term specificity estimation methods are most suitable to be deployed in applications where the answer time is important. In addition to this, given that the methods which combine several modules usually have one which relies on term specificity, it is expected that their results can be improved by selecting the best performing term specificity method available.

The structure of the paper is as follows: In Section 2. we explain how term specificity summarisation methods work ,and briefly explain how the specificity of a term is calculated. Each of the term specificity methods can be applied to different types of tokens, and in Section 3. we explain what type of tokens we use in this paper and their advantages and disadvantages. The methodology used in the evaluation is presented in Section 4. followed by a discussion of the results. The paper finishes with conclusions.

2. Summarisation based on term specificity

Summarisation methods based on term specificity assume that it is possible to determine the importance of a sentence on the basis of the words which constitute it. The most common way to achieve this is to score all the words in the text, and calculate the score of a sentence by adding the scores of the words appearing in it. A summary of the source is produced by extracting the sentences with the highest score until the desired length is achieved and presenting them in the order of their appearance in the original document.

In this section we present four different methods used to compute the relevance score for a word: TF*IDF, a heuristic method commonly used in text summarisation, and TF*RIDF, one of its variants, as well as two methods based on information-theoretic measures of the informativeness of a term - Mutual Information and Information Gain, which are frequently used in text categorisation for term selection, but which, to the best of our knowledge have not been previously studied in relation to text summarisation. In addition, simple term frequency is also used to weight the words even though it is not really a term specificity method.

The task of computing a relevance score for each term in a document can be formalized as follows. Let $d \in D$ be the documents constituting a collection representative of the domain of the source. Let $t \in T$ be the vocabulary of the collection, i.e., the set of all distinct terms appearing in all $d \in D$. Each term t appears in each document with frequency $f_{t,d}$, and the number of documents where t has appeared at least once is $|D_t|$. All term scoring functions under consideration (except term frequency) compute the relevance score $RS(t, d)$ for each t inside each document from t's distribution across $d \in D$.

2.1. Term frequency

Term frequency $f_{t,d}$ is the number of times the term has appeared in the document. The basic assumption behind using it as a relevance score is that frequently appearing terms are most representative of the document while individual rare words are either non-informative or

non-representative of the content of the document. Term frequency has the obvious drawback that frequent terms in a document may well be frequent in general and thus very poor at reflecting the unique content of the document.

2.2. TF*IDF

TF*IDF is a standard term scoring function widely used for various text retrieval tasks which consists of two parts corresponding to two intuitions it embodies: TF for term frequency, the part which favours terms that frequently appear in a particular document, and IDF, which plays down the score of terms that appear in many different documents and thus have poor association with particular documents. We use the most common TF*IDF term scoring schema, computed as:

$$TF * IDF(t,d) = (1 - log f_{t,d}) * log \frac{|D|}{|D_t|} \quad (1)$$

2.3. TF*RIDF

TF*IDF is an ad-hoc method used to score terms, i.e., it does not derive from any mathematical model of the term frequency. Residual IDF, referred to here as RIDF, (Manning and Schütze, 1999, p. 553) is a function which introduces the expected document frequency of a term according to the Poisson model into the estimation of the IDF part of the TF*IDF schema. The formula we used here is:

$$RIDF(t) = IDF - log(1 - p(0; \lambda_t)) \quad (2)$$

where IDF, as in (1) is $log\frac{|D|}{|D_t|}$, and p is the Poisson distribution with parameter λ_t, the average number of occurrences of t per document and $1 - p(0; \lambda_t)$ is the probability of t appearing in a document at least once.

2.4. Mutual Information

Mutual Information (MI) is an information-theoretic measure widely used in statistical NLP for modeling association between two linguistic phenomena (two words, two tags, etc). Mutual Information between a term and a document describes the amount of information the occurrence of the term conveys about the document. We use a global relevance score of term t across for all $d \in D$, computed as the weighted sum of MI(t,d) over all d using the formula proposed in (Mladenic and Grobelnik, 1999):

$$MI(t) = \sum_i P(d_i) * log \frac{P(t|d_i)}{P(t)} \quad (3)$$

2.5. Information Gain

Information Gain (IG), is another well known feature weighting method, introduced into NLP from information theory. IG measures the relevance of term *t* to the document d by computing the difference between the entropies of the document with and without the term. As with MI, we compute a global relevance score (Mladenic and Grobelnik, 1999):

$$IG(t) = P(t) \sum_i P(d_i|t) * log \frac{P(d_i|t)}{P(d_i)} + P(\bar{t}) \sum_i P(d_i|\bar{t}) * log \frac{P(d_i|\bar{t})}{P(d_i)} \quad (4)$$

3. Selection of the tokens

The weighting measures presented in the previous section can be used to score any token obtained from a word. In order to learn how the choice of tokens influences the quality of a summary four different types of token are scored for generating summaries. The four types are:

- *words*: the original form of the word is used without modifications. The drawback of this is that it does not distinguish between different inflections of the word, however, nor does it require any additional computation.
- *truncation* keeps the first 6 characters of a word and converts them to upper case. The drawback is that it considers words such as *comprehension* and *compress* to be derived from the same root *COMPRE*. Its advantage is that it is also very simple to compute.
- *stemming* reduces a word to its stem using a set of predefined rules. The stemmer employed here is the Porter stemmer (Porter, 1980) and relies on a set of rules to remove affixes. Stemming is more accurate than truncation in determining if two words have the same root, but it is still not able to process irregular words.
- *lemmatisation* identifies the lemma of a word. In contrast to truncation and stemming, the result of the lemmatisation process is always a word, which is usually the dictionary look-up form. Lemmatisation requires more resources, but it can deal with irregular words by using lists of exceptions. The lemma of a word is obtained from the lemmatiser included in the WordNet package.

4. Evaluation

In this section we present the corpus used for evaluation, the methodology and the results obtained.

4.1. The corpus used

The evaluation was carried out using part of the corpus described in (Hasler et al., 2003). This part of the corpus contains 147 newswire texts from the Reuters corpus (Rose et al., 2002) with almost 120,000 words in which human annotators marked a total of 30% of the sentences in the texts: 15% as essential, and a further 15% as important.

4.2. Evaluation methodology

Using the human annotation as a gold standard we evaluated all the summarisation methods using precision, recall and f-measure. Because these measures have certain drawbacks we also conducted a small extrinsic evaluation experiment where humans were given 30% summaries and asked to answer 5 questions about the texts.

The evaluation was performed using the CAST environment, which can be used as an evaluation workbench (Orăsan et al., 2003). In this way all the results were obtained using the same preprocessing tools and the same evaluation methodology, which makes the results directly comparable.

4.3. Results

The results of the evaluation are summarised in Table 1. Each term specificity estimation method was applied to all four types of tokens discussed in Section 3., and 15% and 30% extracts were produced. Each method was run with and without a stoplist. *Lead summary* is a summary which contains all the sentences from the beginning of the text up to the limit imposed by the compression rate and it was included here as a baseline.[1] The first number in the table's cells is the average value for the measure corresponding to the column, whereas the second value is the standard deviation for the average value. It should be pointed out that the value of the f-measure presented in the table is not derived from the average values for precision and recall. The value displayed in the table represents the average f-measure.

For the extrinsic evaluation four texts were chosen at random and five questions referring to the central ideas of each text were produced by a linguist. For each text, 30% summaries were generated using the LEAD method, the best performing settings for TF*RIDF and MI. 15 undergraduate students in linguistics received one summary from each text and were asked to answer the questions on the basis of that summary. The results of this experiment are summarised in Table 2.

Method	Correct answers	Total answers	Percent
Lead	57	95	0.60
TF*RIDF	55	80	0.69
MI	53	80	0.66

Table 2: The results of extrinsic evaluation

5. Discussion

The results of the intrinsic evaluation revealed several interesting things. First of all it seems that lead summary is by far the best method for producing 15% summaries from newswire texts. The only method which comes close is TF*RIDF, but the difference is still statistically significant. In light of this, given the additional resources required to compute TF*RIDF, we can conclude that for short summaries of newswire texts, the best approach is to use lead summary.

For longer summaries, lead summary is no longer the best performing method; even the summary based on term frequency performed better in certain circumstances. For 30% summaries, TF*RIDF with stop list obtains the best result when using truncation, good results also being obtained for the other tokens. Mutual Information is another method which performs quite well for 30% summaries, but given the additional computation required, if the time taken to produce the summary is important, TF*RIDF with stop list is a better option. Information Gain performs quite well in text categorisation, but does not seem appropriate for text summarisation.

[1]It should be pointed out that the titles, subtitles and other location information which appears in the newswire texts used here were ignored.

The use of stoplists had a beneficial influence in most cases, but surprisingly there were cases when their use lead to a decrease in the performance, the most notable case being when term frequency is applied to words.

Looking at the results presented in Table 1 it is difficult to identify which token should be scored by a term weighting method in order to obtain the best result. To our surprise the best results are obtained when truncated words are scored by TF*RIDF. This is very convenient from a computational viewpoint, but difficult to justify on the basis of linguistic intuition. An investigation of the results for different types of tokens cannot even suggest which tokens should not be used because they worsen the results, as this depends very much on the scoring method used.

The extrinsic evaluation confirms the fact that TF*RIDF contains most of the important information because the judges who used these summaries were able to answer to the most questions. However, the differences between different methods is not statistically significant. Due to the small size of the experiment the results of the extrinsic evaluation have to be treated with care, larger experiments being necessary to fully validate the results.

6. Conclusions

In this paper, we investigated the influence of different term specificity estimation methods on the results of a term based summarisation method. On the basis of both intrinsic and extrinsic evaluation, the best performing method is TF*RIDF, a variant of the TF*IDF method commonly used in text summarisation. The findings of this paper can be used in two ways. They can be used in applications where the time necessary to produce a summary is important or they can be used in summarisation systems which employ a term weighting module in conjunction with several others in order to boost performance.

7. Acknowledgments

This research was partially funded by the Arts and Humanities Research Board through the "CAST: a Computer Aided Summarisation Tool" project.

8. References

Alonso i Alemany, Laura and Maria Fuentes Fort, 2003. Integrating cohesion and coherence for automatic summarisation. In *Proceedings of EACL2003*. Budapest, Hungary.

Brandow, Ronald, Karl Mitze, and Lisa F. Rau, 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing & Management*, 31(5):675 – 685.

Hasler, Laura, Constantin Orăsan, and Ruslan Mitkov, 2003. Building better corpora for summarisation. In *Proceedings of Corpus Linguistics 2003*. Lancaster, UK.

Manning, Christopher D. and Hinrich Schütze, 1999. *Foundations of statistical natural language processing*. The MIT Press.

Mladenic, Dunja and Marko Grobelnik, 1999. Feature selection for classification based on text hierarchy. In *Conference on Automated Learning and Discovery CONALD-98*.

Method	15%			30%		
	Prec	Recall	F-measure	Prec	Recall	F-measure
Lead summary	20.03 / 0.16	20.08 / 0.16	18.85 / 0.15	45.76 / 0.24	45.84 / 0.24	44.29 / 0.24
Term frequency						
Word	9.93 / 0.10	9.98 / 0.10	8.94 / 0.10	44.47 / 0.19	44.60 / 0.19	43.66 / 0.19
Truncation	9.93 / 0.10	9.98 / 0.10	8.97 / 0.10	44.77 / 0.20	44.90 / 0.20	43.95 / 0.20
Stemming	10.26 / 0.10	10.30 / 0.10	9.27 / 0.10	44.02 / 0.21	44.15 / 0.21	43.09 / 0.20
Lemmatisation	12.05 / 0.11	12.14 / 0.11	11.07 / 0.11	42.06 / 0.20	42.19 / 0.20	41.24 / 0.19
Term frequency with stoplist						
Word	15.68 / 0.13	15.73 / 0.12	14.84 / 0.12	43.48 / 0.21	43.61 / 0.21	42.46 / 0.21
Truncation	15.42 / 0.12	15.46 / 0.12	14.59 / 0.12	44.31 / 0.21	44.44 / 0.21	43.19 / 0.21
Stemming	16.29 / 0.12	16.33 / 0.12	15.44 / 0.12	45.99 / 0.22	46.12 / 0.22	44.93 / 0.22
Lemmatisation	16.75 / 0.13	16.79 / 0.13	15.84 / 0.12	44.29 / 0.21	44.42 / 0.21	43.39 / 0.21
TF * IDF						
Word	13.93 / 0.12	13.97 / 0.12	13.17 / 0.12	46.03 / 0.21	46.13 / 0.21	45.11 / 0.21
Truncation	13.33 / 0.12	13.37 / 0.12	12.57 / 0.12	44.79 / 0.22	44.92 / 0.22	43.63 / 0.22
Stemming	13.82 / 0.13	13.86 / 0.13	12.99 / 0.12	45.39 / 0.21	45.52 / 0.21	44.39 / 0.21
Lemmatisation	16.15 / 0.12	16.20 / 0.12	15.28 / 0.12	45.48 / 0.21	45.57 / 0.21	44.55 / 0.20
TF * IDF with stoplist						
Word	15.29 / 0.13	15.33 / 0.13	14.42 / 0.13	46.11 / 0.22	46.24 / 0.22	45.17 / 0.21
Truncation	13.92 / 0.13	13.97 / 0.13	13.10 / 0.12	45.81 / 0.22	45.94 / 0.22	44.79 / 0.22
Stemming	15.84 / 0.13	15.89 / 0.13	15.04 / 0.12	46.19 / 0.22	46.32 / 0.22	45.23 / 0.21
Lemmatisation	16.48 / 0.12	16.52 / 0.12	15.59 / 0.12	46.64 / 0.21	46.73 / 0.21	45.64 / 0.21
TF * RIDF						
Word	14.11 / 0.13	14.13 / 0.13	13.05 / 0.13	48.60 / 0.23	48.69 / 0.22	47.52 / 0.22
Truncation	13.79 / 0.13	13.82 / 0.13	12.71 / 0.13	47.57 / 0.24	47.64 / 0.24	46.29 / 0.24
Stemming	14.06 / 0.14	14.09 / 0.14	12.92 / 0.13	48.45 / 0.23	48.53 / 0.23	47.28 / 0.23
Lemmatisation	16.31 / 0.14	16.35 / 0.14	15.28 / 0.13	48.64 / 0.22	48.68 / 0.22	47.59 / 0.22
TF * RIDF with stoplist						
Word	17.18 / 0.13	17.22 / 0.13	16.14 / 0.13	48.06 / 0.21	48.19 / 0.21	47.15 / 0.21
Truncation	17.91 / 0.14	17.95 / 0.14	16.93 / 0.13	49.00 / 0.22	49.13 / 0.22	48.05 / 0.21
Stemming	16.57 / 0.13	16.61 / 0.13	15.59 / 0.12	48.24 / 0.21	48.37 / 0.21	47.33 / 0.21
Lemmatisation	17.82 / 0.13	17.87 / 0.13	16.82 / 0.13	47.50 / 0.21	47.63 / 0.21	46.64 / 0.21
Mutual information						
Word	12.60 / 0.12	12.67 / 0.12	11.58 / 0.12	44.44 / 0.22	44.53 / 0.22	43.46 / 0.21
Truncation	12.07 / 0.12	12.13 / 0.12	11.06 / 0.11	45.02 / 0.23	45.11 / 0.23	43.91 / 0.23
Stemming	11.49 / 0.11	11.56 / 0.11	10.64 / 0.10	44.03 / 0.22	44.12 / 0.22	43.13 / 0.21
Lemmatisation	12.95 / 0.12	13.02 / 0.12	11.93 / 0.12	45.02 / 0.23	45.11 / 0.23	43.91 / 0.23
Mutual information with stoplist						
Word	14.26 / 0.12	14.32 / 0.12	13.41 / 0.11	48.05 / 0.24	48.15 / 0.24	46.96 / 0.23
Truncation	13.86 / 0.13	13.92 / 0.13	12.93 / 0.12	48.80 / 0.23	48.91 / 0.23	47.68 / 0.23
Stemming	14.59 / 0.13	14.66 / 0.13	13.46 / 0.13	47.74 / 0.23	47.83 / 0.23	46.66 / 0.23
Lemmatisation	15.60 / 0.14	15.66 / 0.14	14.28 / 0.14	49.00 / 0.24	49.11 / 0.24	47.80 / 0.24
Information gain						
Word	10.59 / 0.12	10.63 / 0.12	9.58 / 0.12	46.51 / 0.22	46.62 / 0.22	45.50 / 0.22
Truncation	10.94 / 0.13	10.99 / 0.13	9.87 / 0.13	47.62 / 0.23	47.73 / 0.23	46.61 / 0.23
Stemming	11.07 / 0.12	11.11 / 0.12	10.09 / 0.11	47.58 / 0.23	47.69 / 0.23	46.53 / 0.23
Lemmatisation	11.65 / 0.13	11.70 / 0.13	10.76 / 0.12	43.21 / 0.23	43.30 / 0.23	42.05 / 0.22
Information gain with stop list						
Word	13.36 / 0.14	13.38 / 0.14	12.19 / 0.13	44.53 / 0.22	44.60 / 0.22	43.41 / 0.22
Truncation	14.46 / 0.13	14.48 / 0.13	13.33 / 0.13	47.00 / 0.22	47.09 / 0.22	45.90 / 0.22
Stemming	14.92 / 0.14	14.94 / 0.14	13.80 / 0.14	45.58 / 0.24	45.67 / 0.24	44.34 / 0.23
Lemmatisation	15.10 / 0.15	15.12 / 0.15	14.02 / 0.14	47.85 / 0.24	47.94 / 0.24	46.54 / 0.24

Table 1: Evaluation of different simple summarisation methods

Orăsan, Constantin, Ruslan Mitkov, and Laura Hasler, 2003. CAST: a Computer-Aided Summarisation Tool. In *Proceedings of EACL2003*. Budapest, Hungary.

Porter, Martin F., 1980. An algorithm for suffix stripping. *Program*, 14(3):130 – 137.

Rose, T. G., M. Stevenson, and M. Whitehead, 2002. The Reuters corpus volume 1 - from yesterday's news to tomorrow's language resources. In *Proceedings of LREC2002*. Las Palmas de Gran Canaria, Spain.