Topic Segmentation Algorithms for Text Summarization and Passage Retrieval: An Exhaustive Evaluation

Gaël Dias*, Elsa Alves*†, José Gabriel Pereira Lopes†

*HULTIG, University of Beira Interior, Covilhã, Portugal
†GLINT, New University of Lisbon, Lisbon, Portugal
{ddg,elsalves}@hultig.di.ubi.pt
gpl@di.fct.unl.pt

Abstract

In order to solve problems of reliability of systems based on lexical repetition and problems of adaptability of language-dependent systems, we present a context-based topic segmentation system based on a new informative similarity measure based on word co-occurrence. In particular, our evaluation with the state-of-the-art in the domain i.e. the c99 and the TextTiling algorithms shows improved results both with and without the identification of multiword units.

Introduction

This paper introduces a new technique for improving access to information dividing lengthy documents into topically coherent sections. This research area is commonly called Topic Segmentation and can be defined as the task of breaking documents into topically coherent multi-paragraph subparts.

In order to provide solutions to access useful information from the ever-growing number of documents on the web, such technologies are crucial as people who search for information are now submerged with unmanageable quantities of texts.

In particular, Topic Segmentation has mainly been used in Passage Retrieval (Cormack et al., 1999) (Yu et al., 2003) and Text Summarization (Barzilay and Elhadad, 1997) (Boguraev and Neff, 2000) (Farzindar and Lapalme, 2004) for the last decade.

However, improvements still need to be made to reliably introduce these techniques into real-word applications. In particular, the systems proposed so far in the literature show three main problems: (1) systems based uniquely on lexical repetition (Hearst, 1994) (Reynar, 1994) (Choi, 2000) show reliability problems as common writing rules prevent from using lexical repetition, (2) systems based on lexical cohesion, using existing linguistic resources that are usually only available for dominating languages like English, French or German, do not apply to less favored languages (Morris and Hirst, 1991) (Kozima, 1993) and (3) systems that need previously existing harvesting training data (Beeferman, Berger and Lafferty, 1997) do not adapt easily to new domains as training data is usually difficult to find or build depending on the domain being tackled.

Instead, our architecture proposes a language-independent unsupervised solution, similar to (Ponte and Croft, 1997), defending that Topic Segmentation should be done "on the fly" on any text thus avoiding the problems of domain/genre/language-dependent systems that need to be tuned each time one of these parameters changes.

For that purpose, our main contributions are twofold. First, we define a new informative similarity measure called InfoSimba that takes into account word co-occurrence and avoids the extra step in the topic identification process as it is the case in (Ponte and Croft, 1997). Second, we clearly pose the problem of word weighting for Topic Segmentation and show that the usual *tf* or *tf.idf* measures (Hearst, 1994) (Reynar, 1994) (Choi, 2000) are not the best heuristics to achieve improved results for this specific task.

In terms of evaluation, (Allan et al., 1998) consider that having a clear evaluation metric is one of the most critical parts of any NLP task. Evaluating a task such as Topic Segmentation consists in determining if the topic shifts are well identified. This can be quite subjective unless you know *a priori* where these boundaries should be placed. To avoid this eventual subjectivity, the evaluation task is usually supervised, by using texts for which we are sure about the topic boundaries. This is achieved by placing in one single document (the one to be segmented) a collection of texts about different issues as in (Choi, 2000) (Ferret, 2002) (Moens and De Busser, 2003). In particular, (Choi, 2000) runs *c99* over a concatenation of text segments, each one extracted from a random selection of the Brown corpus. The Brown corpus consists of 500 texts sampled from 15 different text categories, such as religion, fiction,

and humor. According to many authors (Moens and De Busser, 2003) (Xiang and Hongyuan, 2003), this test set eases the identification of the boundaries as the terms used differ drastically from domain to domain. Instead, (Hearst, 1994) proposes a segmentation algorithm with a different goal: to find subtopic segments i.e. to identify, within a single-topic document, the boundaries of its subparts. A similar experiment is performed by (Xiang and Hongyuan, 2003). In this case, the data set is a set of texts selected from the *Mars* novel. These texts are extracted from different sections and chapters, but we can say they are all about one same issue.

In this paper, we propose an evaluation based on the same idea as (Xiang and Hongyuan, 2003) and build a test set, which is a collection of ten online newspaper articles, covering one same issue: soccer. The choice of this issue is not casual. Independently of the topic of the article (a soccer player being transferred to a different club, a report about a certain game, a championship, etc.), it is usual to find many common words in all texts. As a consequence, we do not favour any algorithm in particular as the test set is a compromise between (Choi, 2000) and (Hearst, 1994) proposals.

The three topic segmentation algorithms, which we evaluate, are the TextTiling (Hearst, 1994), the c99 (Choi, 200) and our InfoSimBa, using three different evaluation metrics: the F-Measure, the P_k estimate (Beeferman, Berger and Lafferty, 1997) and the WindowDiff (Pevzner and Hearst, 2002). The final results of this evaluation show that the InfoSimba obtains improved results both with and without the identification of multiword units compared to the state-of-the-art algorithms.

Word Weighting

Our algorithm is based on the vector space model. The simplest form of the vector space model treats a document (in our case, a sentence or a group of sentences) as a vector whose values correspond to the number of occurrences of the words appearing in the document as in (Hearst, 1994). Although (Hearst, 1994) showed some successful results, we strongly believe that the importance of a word in a document does not only depend on its frequency. According to us, two main factors must be taken into account to define the relevance of a word for the specific task of topic segmentation: its relevance, based on its frequency but also on its inverse document frequency and its distribution across the text. For that purpose, we propose three new heuristics that can be useful to Topic Segmentation: the well-known *tf.idf* measure, the adaptation of the *tf.idf* measure for sentences, the *tf.isf*, and a new density measure, the *dens*.

The tf.idf Score

The idea of the *tf.idf* score (Salton et al., 1975) is to evaluate the importance of a word within a document based on its frequency and its distribution across a collection of documents. The *tf.idf* score is defined in equation 1 where *w* is a word, *d* a document, *tf(w; d)* the number of occurrences of *w* in *d*, |*d*| the number of words in *d*, *df(w)* the number of documents in which the word *w* occurs and *N* the size of the collection of documents.

$$tf.idf(w,d) = \frac{tf(w;d)}{|d|} \times \ln \frac{N}{df(w)} \quad (1)$$

However, not all relevant words in a document are useful for Topic Segmentation. For instance, relevant words appearing in all sentences will be of no help to segment the text into topics. For that purpose, we extend the idea of the *tf.idf* to sentences, the *tf.isf*.

The tf.isf Score

The basic idea of the *tf.isf* score is to evaluate each word in terms of its distribution over the document. Indeed, it is obvious that words occurring in many sentences within a document may not be useful for topic segmentation purposes. So, we define the *tf.isf* to evaluate the importance of a word within a document based on its frequency within a given sentence and its distribution across all the sentences within the document. The *tf.isf* score is defined in equation 2 where *w* is a word, *s* a sentence, *stf(w; s)* the number of occurrences of *w* in *s*, |*s*| the number of words in *s*, *sf(w)* the number of sentences in which the word *w* occurs and *Ns* the number of sentences within the document.

$$tf.isf(w) = \frac{stf(w;s)}{|s|} \times \ln \frac{Ns}{sf(w)} \quad (2)$$

As a result, a word occurring in all sentences of the document will have an inverse sentence frequency equal to 0, giving it no chance to be a relevant word for Topic Segmentation. On the opposite, a word which occurs very often in one sentence, but in very few other sentences, will have a high inverse sentence frequency as well as a high sentence term frequency and thus a high *tf.isf* score. Consequently, it will be a strong candidate for being a relevant word within the document for the specific task of Topic Segmentation. However, we can push even further our idea of word distribution. Indeed, a word *w* occurring 3 times in 3 different sentences may not have the same importance in all cases. Let's exemplify. If the 3 sentences are consecutive, the word *w* will have a strong influence on what is said in this specific region of the text. On the opposite, it will not be the case if the word *w* occurs in the first sentence, in the middle sentence and then in the last sentence. It is clear that we must take into account this phenomenon. For that purpose, we propose a new density measure that calculates the density of each word in a document.

The Density Measure

The basic idea of the word density measure is to evaluate the dispersion of a word within a document. If a word *w* appears in consecutive or near consecutive sentences it will have a strong influence on what is said in this specific region of the text whereas if it occurs in distant sentences, its importance will be negligible. In order to evaluate the word density, we propose a measure based on the distance of all consecutive occurrences of the word in the document. We call this measure *dens(.,.)* defined in equation 3.

$$dens(w,d) = \sum_{k=1}^{|w|-1} \frac{1}{\ln\left(dist\left(occur(k), occur(k+1)\right)+e\right)} \qquad (3)$$

For any given word *w*, its density *dens(w,d)* in document *d*, is calculated from all the distances between all its occurrences, |*w*|. So, *occur(k)* and *occur(k+1)* respectively represent the positions in the text of two consecutive occurrences of the word *w* and *dist(occur(k), occur(k+1))* calculates the distance separating them in terms of words. Thus, by summing their inverse distances, we get a density function that gives higher scores to highly dense words. As a result, a word, which occurrences appear close to one another, will show small distances and as a result a high density. On the opposite, a word, which occurrences appear far from each other, will show high distances and as a result a small word density.

The Weighting Score

The weighting score of any word in a document can be directly derived from the previous three heuristics. For that purpose, we use an linear interpolation of all heuristics as shown in equation 4, where each individual score is normalized following the ratio paradigm, α, β, γ are constants and ⊗ is the product or the sum function.

$$weight(w,d) = \alpha.\left\|tf.idf(w,d)\right\| \otimes \beta.\left\|tf.isf(w)\right\| \otimes \gamma.\left\|dens(w,d)\right\| \qquad (4)$$

However, this computation is not made for all the words in text. In fact, we ignore the most frequent words which appear in the text. We started to do it with the intention of decrease the processing time, but the exclusion of these words improves the performance of all the algorithms. These, so called, most common words or stop words, are detected by the simple fact of being the most frequent words over the text to be segmented. This detection is absolutely automatic and dynamic. After performing some tests, we concluded that 10 is the ideal size for this list.

Once all words in the document to segment have been evaluated in terms of relevance and distribution, the next step of the application is to determine similarities between a focus sentence and its neighboring groups of sentences.

Similarity between Sentences

There are a number of ways to compute the similarity between two documents, in our case, between a sentence and a group of sentences. Theoretically, a similarity measure can be defined as follows. Suppose that $Xi = (X_{i1}, X_{i2}, X_{i3}, \ldots, X_{ip})$ is a row vector of observations on *p* variables associated with a label *i*. The similarity between two units, *i* and *j*, is defined as $S_{ij} = f(X_i, X_j)$ where *f* is some function of the observed values. In the context of our work, the application of a similarity measure is straightforward. Indeed, X_i may be regarded as the focus sentence and X_j as a specific block of *k* sentences, each one being represented as *p*-dimension vectors, where *p* is the number of different words within the document and where X_{ib} represents the weighting score of the b^{th} word in the document also appearing in the focus sentence X_i. Our goal here is to find the appropriate *f* function that will accurately evaluate the similarity between the focus sentence and the blocks of *k* sentences. For that purpose, we introduce a new informative similarity measure called the InfoSimba.

Most of the NLP applications have been applying the cosine similarity measure. However, when applying the cosine between two documents, only the identical indexes of the row vectors X_i and X_j will be taken into account i.e. if both documents do not have words in common, they will not be similar at all and will receive a cosine value of 0. However, this is not tolerable. Indeed, it is clear that both sentences (1) and (2) are similar although they do not share any word:

(1) *Ronaldo defeated the goalkeeper once more*.
(2) *Real Madrid striker scored again*.

A much more interesting research direction is proposed by (Ponte and Croft, 1997) who propose a Topic Segmentation technique based on the Local Content Analysis, allowing substituting each sentence with words and phrases related to it. Our methodology is based on this same idea but differs from it as the word co-occurrence information is directly embedded in the similarity measure thus avoiding an extra-step in topic boundaries discovery. For that purpose, we propose a new informative similarity measure that includes in its definition the Equivalence Index Association Measure (*EI*) proposed by (Muller et al., 1997). It is defined in equation 5.

$$EI(w_1, w_2) = p(w_1 \mid w_2) \times p(w_2 \mid w_1) = \frac{f(w_1, w_2)^2}{f(w_1) \times f(w_2)} \qquad (5)$$

The Equivalence Index between words w_1 and w_2 is calculated within a context window in order to determine $f(w_1, w_2)$ and from a collection of documents so that we can evaluate the degree of cohesiveness between two words outside the context of the document. This collection can be thought as the overall web, from which we are able to infer with maximum reliability the "true" co-occurrence between two words. So, the basic idea of our informative similarity measure is to integrate into the cosine measure

the word co-occurrence factor inferred from a collection of documents with the *EI* association measure. This can be done straightforwardly as defined in equation 6.

$$S_{ij} = \text{infosimba}\left(X_i, X_j\right) = \frac{A_{ij}}{B_i \times B_j + A_{ij}} \quad (6)$$

where

$$A_{ij} = \sum_{k=1}^{p}\sum_{l=1}^{p} X_{ik} \times X_{jl} \times EI\left(w_{ik}, w_{jl}\right) \quad \text{and}$$

$$\forall i, B_i = \sqrt{\sum_{k=1}^{p}\sum_{l=1}^{p} X_{ik} \times X_{il} \times EI\left(w_{ik}, w_{il}\right)}$$

and any X_{zv} corresponds to the word weighting factor *weight*(w_{zv},*d*) and *EI*(w_{ik},w_{jl}) is the Equivalence Index value between w_{ik}, the word that indexes the word context vector i at position k and w_{jl}, the word that indexes the word context vector j at position l. In fact, the informative similarity measure can simply be explained as follows. For example, *Real_Madrid_striker*[1] would give rise to the sum of 6 products i.e. *Real_Madrid_striker* with *Ronaldo*, *Real_Madrid_striker* with *defeated* and so on and so forth. As a consequence, sentence (1) and (2) would show a high similarity as *Real_Madrid_striker* is related to *Ronaldo*.

Topic Boundaries Detection

Different methodologies have been proposed to place subtopic boundaries between dissimilar blocks (Kozima, 1993) (Hearst, 1994) (Ponte and Croft, 1997) (Beeferman, Berger and Lafferty, 1997) (Stokes and Carthy, 2002). For that purpose, we propose a new methodology based on ideas expressed by different research. Taking as reference the idea of (Ponte and Croft, 1997) who take into account the preceding and the following contexts of a segment, we calculate the informative similarity of each sentence in the corpus with its surrounding pieces of texts i.e. its previous block of k sentences and its next block of k sentences. The idea is to know whether the focus sentence is more similar to the preceding block of sentences or to the following block of sentences. In order to evaluate this preference in an elegant way, we propose a score for each sentence in the text in the same way (Beeferman, Berger and Lafferty, 1997) compare short and long-range models. Our preference score (*ps*) is defined in equation 7.

$$ps(S_i) = \ln \frac{\text{infosimba}(S_i, X_{i-1})}{\text{infosimba}(S_i, X_{i+1})} \quad (7)$$

So, if $ps(S_i)$ is positive, it means that the focus sentence S_i is more similar to the previous block of sentences, X_{i-1}. Conversely, if $ps(S_i)$ is negative, it means that the focus sentence S_i is more similar to the following block of sentences, X_{i+1}. In particular, when $ps(S_i)$ is near 0, it means that the focus sentence X_i is similar to both blocks and so in the continuity of a topic. In order to better understand the variation of the *ps(.)* score, each time its value goes from positive to negative between two consecutive sentences, there exits a topic shift. We will call this phenomenon a downhill. However, not all downhills identify the presence of a new topic in the text. Indeed, only deeper ones must be taken into account. In order to automatically identify these downhills, and as a consequence the topic shifts, we adapt the algorithm proposed by (Hearst, 1994). So, we propose a threshold that is a function of the average and the standard deviation of the downhills depths. A downhill is simply defined in equation 8 whenever the value of the *ps(.)* score goes from positive to negative between two sentences S_i and S_{i+1}.

$$downhill\left(S_i, S_{i+1}\right) = ps\left(S_i\right) - ps\left(S_{i+1}\right) \quad (8)$$

Once all downhills have been calculated, their mean $\bar{x}$ and standard deviation σ are evaluated. The topic boundaries are then elected if they satisfy the constraint expressed in equation 9 where c is a constant to be tuned.

$$downhill\left(S_i, S_{i+1}\right) \geq \bar{x} + c\sigma \quad (9)$$

Evaluation: Results and Discussion

The Benchmark

We built our own benchmark based on real-world texts taken from the web from a single domain, Soccer. In fact, we automatically gathered 100 articles of approximatively 100 words. We built 10 test corpora, by choosing randomly 10 articles from our database of 100 articles leading to 10 texts of around 1000 words-long.

Evaluation Metrics

In order to evaluate the performance of the compared systems, we used three distinct metrics: the F-measure, the P_k estimate (Beeferman, Berger and Lafferty, 1997) and the WindowDiff measure (Pevzner and Hearst, 2002).

Multiword Units

As (Ferret, 2002) showed that improved results can be obtained by the identification of collocations, we also proposed a set of experiments using both words and multiword units as basic textual units. In order to semantically enrich the texts, we used the SENTA Software proposed by (Dias, Guilloré and Lopes, 1999)[2].

Evaluation Scheme

In order to be as complete as possible, we ran the c99 algorithm, the TextTiling algorithm and our algorithm on our benchmark from which multiword units have been identified. Our evaluation scheme gave rise to 14

[1] In this example, the multiword unit Real_Madrid_Striker would be identified by the multiword extractor SENTA (Dias, Guilloré and Lopes, 1999).

[2] http://senta.di.ubi.pt

experiments for which the F-measure, the P_k estimate and the WindowDiff measure have been evaluated. All results are illustrated in Table 1.

The first result is that the c99 algorithm is the one that worst performs over our test corpus. This goes against (Choi, 2000)'s evaluation that evidences improved results when compared to the TextTiling algorithm over the c99 corpus. This result clearly shows that the c99 can not be taken as a gold standard for Topic Segmentation evaluation schemes as it has been done in many works. The reason why the TextTiling algorithm performs better than the c99 on our benchmark is the fact that (Hearst, 1994) uses the appearance of new lexical units as a clue for topic boundary detection whereas (Choi, 2000) relies more deeply on lexical repetition which drastically penalizes the topic boundary detection process.

Secondly, we performed several tests in order to evaluate the real value of each measure used by our algorithm itself. In fact, we were interested to verify what role the measures were having in the results. To do so, we combined the three different metrics (*tf.idf*, *tf.isf* and *dens*) by setting the constant values α, β and γ equals to 0, or 1, as we wanted to exclude, or include, its value. Curiously, the best results were not achieved by using the three measures at the same time. In fact, in the case where SENTA is not applied, the overall architecture works better with the *tf.idf* measure alone than combined with the other measures going towards the results shown by (Hearst, 1994). It is also true that the difference is not very relevant but it is a reality, which lead us to analyze the results after applying the SENTA system where we can see that, with multiword unit detection, it becomes necessary to use one of the local measures to achieve the best results, being the *dens(.,.)* measure the one which evidences best results.

To answer some questions about which metric performs best, *tf.idf* or *tf* alone, we tested our algorithm for both cases. The numbers speak for themselves. Definitely, term frequency alone is not enough contrarily to what (Hearst, 1994) claims.

Even more interesting were the results returned by the tests where we ignored all three measures, leaving only the Equivalence Index of the InfoSimba similarity measure taking into account lexical cohesion. Over the texts without multiword units, the use of the three measures, standing alone or combined, return worse results than if we ignore them. Again, the distance between the results is not significant, but it is relevant to prove the importance of the co-occurrence between the pairs of words. However, as the use of multiword units already implies this co-occurrence, as mentioned before, we get better result by adding the combination of the *tf.idf* and *dens* measures.

Facing these results, we thought that it would be interesting to see what would happen if we did not use the InfoSimba but the cosine measure instead. As it is shown in Table 1, the results are far from good and clearly evidence the contribution of the InfoSimba.

When we intended to analyze the weight of the *most common words*, we verified that their inclusion in the evaluation decreased significantly the performance of the algorithm as shown in Table 1 also.

After this, we ran our algorithm using the multiplication operator in the calculus of each word weight. As predicted, the results were worse, being justified by the fact that, by multiplying, relevant words found by one of the statistics could be depreciated by other, while using the sum operator, all additional information about the relevancy of each other is, actually, added.

Conclusion

In this paper, we proposed a language-independent Topic Segmentation system based on word co-occurrence, which avoids the accessibility to existing linguistic resources and does not rely on lexical repetition. To our point of view, our main contribution to the field are the new weighting scheme and the definition of a new similarity measure, the informative similarity measure that proposes a mathematical model that deals with the word co-occurrence factor and avoids an extra step in the boundary detection compared to the solution introduced by (Ponte and Croft, 1997). In order to evaluate our system, we compared it to the state-of-the-art in the domain i.e. the c99 (Choi, 2000) and TextTiling (Hearst, 1994) algorithms on a real-world web corpus and measured its performance based on three metrics: the classical F-measure, the P_k estimate (Beeferman, Berger and Lafferty, 1997) and the WindowDiff measure (Pevzner and Hearst, 2002). In particular, our system demonstrated at most 33% improvements over the c99 algorithm and 24% over the TextTiling algorithm in terms of F-measure. In terms of the P_k estimate and the WindowDiff measure, our system also showed better results except for one case where the WindowDiff over-evalutes near misses. One important conclusion of our evaluation is the fact that none of the three evaluation metrics is satisfactory and work still need to be done in this area.

References

Allan, J., Carbonell, J., Doddington, G., Yamron J. and Yang, Y. 1998. *Topic detection and tracking pilot study: Final report*. In Proc. of the DARPA Broadcast News.

Barzilay, R., and Elhadad, M. 1997. *Using lexical chains for text summarization*. In Proc. of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL.

Beeferman, D., Berger, A. and Lafferty, J. 1997. *Text Segmentation using Exponential Models*. In Proc. of the Second Conference on Empirical Methods in Natural Language Processing, 35-46.

Boguraev, B. and Neff, M. 2000. *Discourse Segmentation in Aid of Document Summarization*. In Proc. of Hawaii International Conference on System Sciences (HICSS- 33), Minitrack on Digital Documents Understanding.

Choi, F.Y.Y. 2000. *Advances in Domain Independent Linear Text Segmentation*. In Proceedings of NAACL'00.

Cormack, G.V., Clarke, C.L.A., Kisman, D.I.E. and Palmer, C.R. 1999. *Fast Automatic Passage Ranking. MultiText Experiments for TREC-8*. In Proc. of TREC-8. 735-742.

Dias, G., Guilloré, S. and Lopes, J.G.P. 1999. *Language Independent Automatic Acquisition of Rigid Multiword Units from Unrestricted Text corpora*. In Proc. of 6ème Conférence Annuelle sur le Traitement Automatique des Langues Naturelles, July 12-17. 333-339.

Farzindar, A. and Lapalme, G. 2004. *Legal Text Summarization by Exploration of the Thematic Structures and Argumentative Roles*. In Workshop on Text Summarization Branches Out, ACL 2004.

Ferret, O. 2002. *Using Collocations for Topic Segmentation and Link Detection*. In Proc. of COLING 2002, 19th International Conference on Computational Linguistics.

Hearst, M. 1994. *Multi-Paragraph Segmentation of Expository Text*. In Proc. of the 32nd Meeting of the Association for Computational Linguistics.

Kozima, H. 1993. *Text Segmentation Based on Similarity between Words*. In Proc. of the 31th Annual Meeting of the Association for Computational Linguistics, 286-288.

Moens, M-F. and De Busser, R. 2003. *Generic Topic Segmentation of Document Texts*. In Proc. of the 24th annual international ACM SIGIR conference on Documentation. San Francisco, USA. 117-124.

Morris, J. and Hirst, G.. 1991. *Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text*. Computational Linguistics 17(1): 21-43. 1991.

Muller, C., Polanco, X., Royauté, J. and Toussaint, Y. 1997. *Acquisition et Structuration des Connaissances en Corpus: Eléments Méthodologiques*. Technical Report RR-3198, Inria.

Pevzner, L. and Hearst, M. 2002. *A Critique and Improvement of an Evaluation Metric for Text Segmentation*. Computational Linguistics, 28 (1), March. 19-36.

Ponte, J.M. and Croft, W.B. 1997. *Text Segmentation by Topic*. In Proc. of the First European Conference on Research and Advanced Technology for Digitial Libraries.120-129.

Reynar, J.C. 1994. *An Automatic Method of Finding Topic Boundaries*. In Proc. of the 32th Annual Meeting of the Association for Computational Linguistics.

Salton, G., Yang, C.S. and Yu, C.T. 1975. *A Theory of Term Importance in Automatic Text Analysis*. In Amer. Soc. Inf. Science 26, 1, 33-44.

Stokes, N., Carthy, J. and Smeaton, A.F. 2002. *Segmenting Broadcast News Streams Using Lexical Chains*. In Proc. of 1st Starting AI Researchers Symposium (STAIRS 2002), volume 1, 145-154.

Xiang, J. and Hongyuan, Z. 2003. *Domain-independent Text Segmentation Using Anisotropic Diffusion and Dynamic Programming*. In Proc. of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 322-329.

Yu, S., Cai, D., Wen, J-R., and Ma, W.Y. 2003. Improving Pseudo-Relevance Feedback in Web Information Retrieval Using Web Page Segmentation. In Proceedings of *WWW2003*, May 20-24, 2003, Budapest, Hungary. ACM 1-58113-680-3/03/0005.

Table 1: Results of our experiments.

Without SENTA						With SENTA				
Prec	**Rec**	**F-Mea**	**Pk**	**WD**		**Prec**	**Rec**	**F-Mea**	**Pk**	**WD**
0,65	0,80	0,72	0,21	0,25	$\alpha = \beta = \gamma = 0$	0,66	0,81	0,72	0,22	0,25
0,66	0,78	0,71	0,21	0,26	$\alpha = 1, \beta = \gamma = 0$	0,67	0,84	0,74	0,19	0,26
0,49	0,55	0,51	0,35	0,37	$\beta = 1, \alpha = \gamma = 0$	0,45	0,52	0,48	0,34	0,38
0,58	0,73	0,65	0,26	0,30	$\gamma = 1, \alpha = \beta = 0$	0,65	0,83	0,73	0,21	0,30
0,66	0,77	0,71	0,22	0,25	$\alpha = \beta = 1, \gamma = 0$	0,66	0,83	0,73	0,19	0,27
0,64	0,75	0,69	0,22	0,26	$\alpha = \gamma = 1, \beta = 0$	0,68	0,87	0,76	0,17	0,25
0,54	0,56	0,54	0,34	0,35	$\alpha = 0, \beta = \gamma = 1$	0,62	0,77	0,68	0,24	0,29
0,64	0,76	0,69	0,22	0,26	$\alpha = \beta = \gamma = 1$	0,66	0,83	0,74	0,19	0,27
0,45	0,48	0,46	0,37	0,39	Term Frequency	0,41	0,44	0,42	0,38	0,42
0,17	0,12	0,14	0,53	0,46	Cosine Similarity	0,11	0,08	0,09	0,55	0,48
0,44	0,36	0,33	0,41	0,37	With Most Common Words	0,51	0,51	0,51	0,33	0,35
0,17	0,13	0,13	0,51	0,43	Multiplication	0,17	0,16	0,16	0,53	0,46
0,50	0,40	0,44	0,31	0,37	c99	0,50	0,39	0,44	0,31	0,36
0,45	0,50	0,47	0,33	0,30	TextTiling	0,55	0,52	0,53	0,24	0,26