Discursive Sentence Compression

Alejandro Molina[1], Juan-Manuel Torres-Moreno[1], Eric SanJuan[1],
Iria da Cunha[2], and Gerardo Eugenio Sierra Martínez[3]

[1] LIA-Université d'Avignon
[2] IULA-Universitat Pompeu Fabra
[3] GIL-Instituto de Ingeniería UNAM

Abstract. This paper presents a method for automatic summarization by deleting intra-sentence discourse segments. First, each sentence is divided into elementary discourse units and, then, less informative segments are deleted. To analyze the results, we have set up an annotation campaign, thanks to which we have found interesting aspects regarding the elimination of discourse segments as an alternative to sentence compression task. Results show that the degree of disagreement in determining the optimal compressed sentence is high and increases with the complexity of the sentence. However, there is some agreement on the decision to delete discourse segments. The informativeness of each segment is calculated using textual energy, a method that has shown good results in automatic summarization.

1 Introduction

Previous studies in automatic summarization have proposed to generate summaries by extracting certain sentences of a given document; i.e., an *extraction summary* [1]. Nevertheless, an *abstract*, as defined in [2], is by far the most concrete and most recognized document summarization method.

Today, automatic summarization approaches have improved to the point that they are able to identify, with remarkable precision, the sentences that contain the most essential information for any given text. However, high-scored sentences could contain a great amount of irrelevant information. Hence, a finer analysis is needed to prune the superfluous information while retaining only the relevant one [3]. Sentence compression shall produce grammatical condensed sentences that preserve important content and it represents a valuable resource for automatic summarization systems. Indeed, some authors argue that this task could be a first step towards abstract generation [4].

This work presents a sentence compression approach for summarization. First, sentences are segmented using a discourse segmenter and, then, a compressed text is generated erasing the less informative segments. Statistical methods allow to determinate segment's informativeness and grammaticality.

The rest of the paper is organised as it follows. In section 2, the main concepts of sentence compression are covered. In section 3, the discourse segmentation is presented. Then the description of how to score the informativeness

A. Gelbukh (Ed.): CICLing 2013, Part II, LNCS 7817, pp. 394–407, 2013.

of the segments is presented in section 4 and its grammaticaility in section 5. The experimental protocol and the results are shown in section 6 and section 7 respectively. Finally, section 8 presents our conclusions and future work.

2 Summarization by Compression

2.1 Classic Sentence Compression

In [4], the sentence compression task is defined as: "Consider an input sentence as a sequence of n words $W = (w_1, w_2, ..., w_n)$. An algorithm may drop any subset of these words. The words that remain (order unchanged) form a compression". The authors included a standard corpus for sentence compression. Later, [5] confirmed that results could be interesting for text summarization and [6] used a similar approach for speech summarization. In [7], a sentence compression corpus, was annotated by humans considering the context. Nonetheless, the criteria used to elicit the compressions remain quite artificial for summarization. The autors asked the annotators to delete individual words from each sentence, but humans also tend to delete long phrases in an abstract. In all of these works it should be noticed a major drawback associated to individual words deletion: deleting individual words could be very risky in terms of grammar, and too poor in terms of compression rate. One single word deletion can seriously affect the sentence, for instance, erasing a verb or a negation.

2.2 More Recent Approaches in Sentence Compression

Recent studies have found outstanding results using clauses or discourse structures, instead of isolated words. An algorithm, proposed in [8], divides sentences into clauses prior to any elimination. Although the results of this last work are good in general, in some cases the main subject of the sentence is removed. The authors attempted to solve this issue by including features in a machine learning approach [9].

Discourse chunking [10] is an alternative to discourse parsing, thereby, showing a direct application to sentence compression as shown in [11]. The authors of these last two works argued that, while discourse parsing at document level is a significant challenge, discourse chunking at sentence level could present an alternative in human languages with limited language processing tools. In addition, some sentence-level discourse models have shown accuracies comparable to human performance [12].

3 Discourse Segmentation

3.1 Sentence Level Discourse Segmentation

In this work, we use a sentence-level discourse segmentation approach. Formally, "Discourse segmentation is the process of decomposing discourse into Elementary Discourse Units (EDUs), which may be simple sentences or clauses in a

complex sentence, and from which discourse trees are constructed" [13]. The first step of discourse parsing is discourse segmentation (the next steps are detection of rhetorical relations and building of the discourse tree). However, we can consider segmentation at the sentence level in order to identify segments to be eliminated in the sentence compression task. The decomposition of a sentence into EDUs using only local information is called *intra-sentence discourse segmentation*. Today, automatic discourse segmentation systems exist for several languages such as English [13], Brazilian Portuguese [14], Spanish [15] and French [16].

3.2 Compression Candidates Generation

In this work we propose to generate compression candidates by deletion of some discourse segments from the original sentence. Let be a sentence S the sequence of its k discourse segments: $S = (s_1, s_2, ..., s_k)$. A candidate, CC_i, is a subsequence of S that preserves the original order of the segments. The original sentence always forms a candidate, i.e., $CC_0 = S$, this is convenient because sometimes there is no shorter grammatical version of the sentence, especially in short sentences that conform one single EDU. Since we do not consider the empty subsequence as a candidate, there are $2^k - 1$ candidates.

3.3 The DiSeg Discourse Segmenter

The discourse segmenter used in our experiments, DiSeg, is described in [15] and is based on the Rhetorical Structure Theory [17]. This system detects discourse boundaries in sentences. First, a text is pre-processed with sentence segmentation, POS tagging and shallow parsing modules using the Freeling toolkit[18]. Then, an XML file is generated with discourse marker annotations. Finally, several rules are applied to this file. The rules are based on: discourse markers, as "while" (*mientras que*), "although" (*aunque*) or "that is" (*es decir*), which usually mark the relations of CONTRAST, CONCESSION and REFORMULATION, respectively; conjunctions, such as "and" (*y*) or "but" (*pero*); adverbs, as "anyway" (*de todas maneras*); verbal forms, as gerunds, finite verbs, etc.

3.4 Adapting DiSeg to Sentence Compression: The CoSeg Segmenter

We have adapted the discourse segmenter DiSeg for the sentence compression task simply by modifying its original rules in order to ease the definition of EDUs. While in DiSeg it is mandatory that every EDU contains a principal verb, in CoSeg, a segment could not have any verb. In CoSeg, if a fragment contains a discourse marker it must be segmented. We also consider that punctuation marks, as parenthesis, comas or dashes, are natural boundaries in sentences. Afterall, the final goal is to create a sentence compression system based on this adapted version of DiSeg.

4 Informativeness of Discourse Segments

4.1 The Textual Energy

$$S = \begin{pmatrix} s_1^1 & s_1^2 & \cdots & s_1^N \\ s_2^1 & s_2^2 & \cdots & s_2^N \\ \vdots & \vdots & \ddots & \vdots \\ s_P^1 & s_P^2 & \cdots & s_P^N \end{pmatrix} ; \quad s_\mu^i = \begin{cases} TF_i \text{ if word } i \text{ is present in segment } \mu \\ 0 \quad \text{elsewhere} \end{cases} \tag{1}$$

Textual energy is a similarity measure used in several NLP tasks: automatic summarization [19], topic segmentation [20] and text clustering [21]. In this method, words and sentences are taken as a magnetic system composed of spins (words coded as 1's and 0's). In its original description [22], the minimal unit of processing is the sentence and the main idea is to rank all of the sentences in a text. In this work, we use the textual energy for the evaluation of discourse segments. First, documents are pre-treated with classical algorithms like filtering and lemmatisation to reduce the dimensionality (see [23] for details). Then, a *bag of words*, representing the segments, produces the matrix $S_{[P\times N]}$ (1) of word frequencies/absences consisting of $\mu = 1, \cdots, P$ segments (rows) and the vocabulary of $i = 1, \cdots, N$ terms (columns).

Let us consider the segments as sets σ of words. These sets constitute the vertices of a graph like that of the Figure 1. We can draw an edge between two of the vertices σ_μ, σ_ν every time they share at least a word in common $\sigma_\mu \cap \sigma_\nu \neq \emptyset$. We obtain the graph $I(S)$ from intersection of the segments. We evaluate these pairs $\{\sigma_1, \sigma_2\}$, which we call edges, by the exact number $|\sigma_1 \cap \sigma_2|$ of words that share the two connected vertices. Finally, we add to each vertex σ an edge of reflexivity $\{\sigma\}$ valued by the cardinal $|\sigma|$ de σ. This valued intersection graph is isomorphic with the adjacency graph $G(S \times S^T)$ of the square matrix $S \times S^T$. In fact, $G(S \times S^T)$ contains P vertices. There is an edge between two vertices μ, ν if and only if $[S \times S^T]_{\mu,\nu} > 0$. If it is the case, this edge is valued by $[S \times S^T]_{\mu,\nu}$ and this value corresponds to the number of words in common between the segments μ and ν. Each vertex μ is balanced by $[S \times S^T]_{\mu,\mu}$, which corresponds to the addition of an edge of reflexivity. It results that the matrix of Textual Energy E is the adjacency matrix of the graph $G(S \times S^T)^2$. The textual energy of segments interaction can be expressed by (2).

$$E = -\frac{1}{2} S \times (S^T \times S) \times S^T = -\frac{1}{2}(S \times S^T)^2 \tag{2}$$

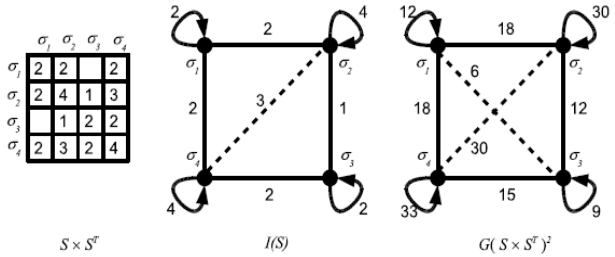

Fig. 1. Graph from the matrix of energy

4.2 Segment Energy

Given that, the sum of the row j in the energy matrix (2) gives the lexical link strengths of segment j, therefore we are able to determine how relevant segment j is in the text. Textual energy matrix connects segments having common words, as well as segments sharing the same neighbourhood but not necessarily identical vocabulary.

Figures 2 and 3 show a text extracted from our corpus. Each row corresponds to a segment while first and second columns correspond to the energy values considering the original sentence energy and individual segment energy. The gray tonality exhibits the degree of informativeness of the segments considering the whole text context: darker segments are the less informative. Bottoms of the figures show the density plot of energy values. Table 1 shows the approximate translations for both segmenters with the original sentences numbered.

5 Grammaticality of Discourse Segments

5.1 Scoring Discourse Segments with Language Models

Statistical language modeling [23] is a technique widely used to assign a probability to a sequence of words. The probabilities in a Language Model (LM) are estimated counting sequences from a corpus. Even though we will never be able to obtain enough data to compute the statistics for all possible sentences, we can base our estimations using large corpora and interpolation methods. In the experiments we use a big corpus with 1T words (LDC Catalog No.: LDC2009T25) to obtain the sequence counts and a LM interpolation based on Jelinek-Mercer smoothing [24]. In a LM, the maximum likelihood estimate of a sequence is interpolated with the smoothed lower-order distribution. We use the Language Modeling Toolkit SRILM [25] to score the segment likelihood probability. We assume that good compression candidates must have a high probability as sequences in a LM.

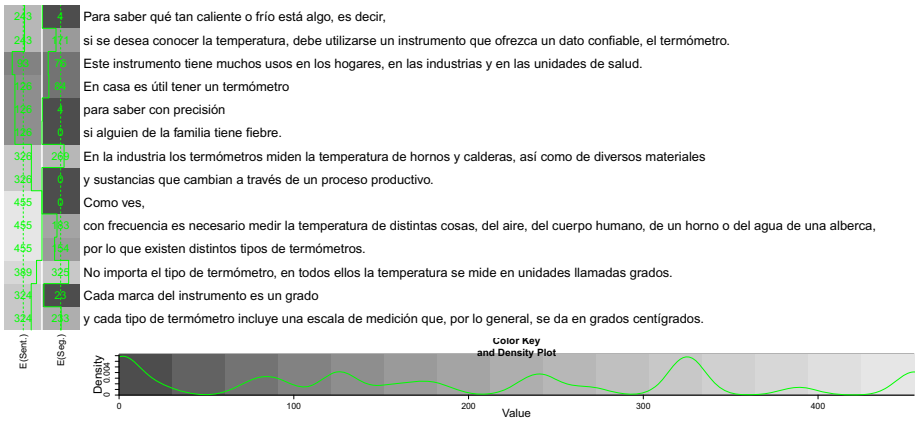

Fig. 2. Textual energy values for DiSeg segments in a text

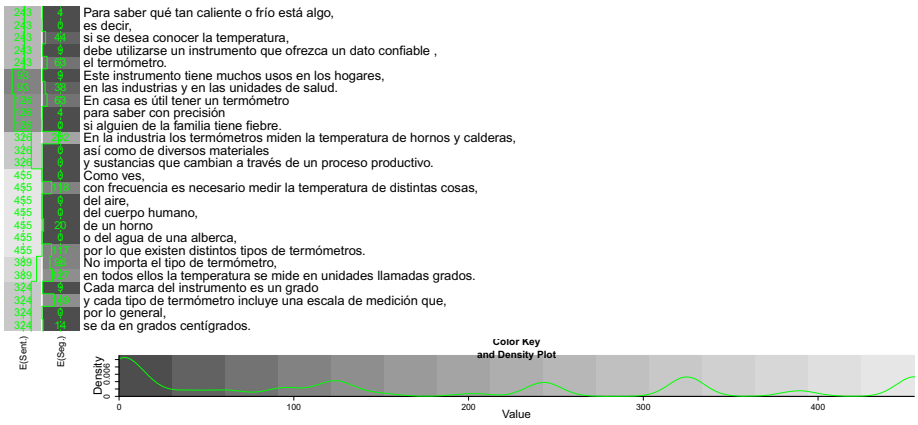

Fig. 3. Textual energy values for Coseg segments in a text

Table 1. Approximate segmentations translated for both segmenters

DiSeg	CoSeg
1. To find out how hot or cold something is, that means,	**1.** To find out how hot or cold something is,
if you want to know the temperature, use an instrument that provides reliable data, the thermometer.	that means,
2. This instrument has many uses at home, industries and care units.	if you want to know the temperature,
3. Having a thermometer at home is useful	use an instrument that provides reliable data,
to know with precision	the thermometer.
if a family member has fever.	**2.** This instrument has many uses at home,
4. In the industry thermometers measure the temperature of furnaces and boilers, as well as various materials	industries and care units.
and substances that change through a productive process.	**3.** Having a thermometer at home is useful
5. As you can see,	to know with precision
it is often necessary to measure the temperature of different things, the air, the human body, a furnace or water of a pool,	if a family member has fever.
that's why there are different types of thermometers.	**4.** In the industry thermometers measure the temperature of furnaces and boilers,
6. No matter what type of thermometer, in every case the temperature is measured in units called degrees.	as well as various materials
7. Each mark in the thermometer is a degree	and substances that change through a productive process.
and each type of thermometer includes a measuring scale, which in general, is given in degrees Celsius.	**5.** As you can see,
	it is often necessary to measure the temperature of different things,
	the air,
	the human body,
	a furnace
	or water of a pool,
	that's why there are different types of thermometers.
	6. No matter what type of thermometer,
	in every case the temperature is measured in units called degrees.
	7. Each mark in the thermometer is a degree
	and each type of thermometer includes a measuring scale, which
	in general,
	is given in degrees Celsius.

6 Experiments

6.1 To Delete or Not to Delete, That Is the Task

We have set up a campaign of text annotation with non-expert volunteers (a citizen science project). First, we have chosen 30 short texts. Then, each text was segmented twice: one using DiSeg and the other one with CoSeg. We asked human annotators to chose which segments must be preserved to form an abstract, following the criteria in section 6.2 hereafter. Figure 4 shows the interface used during the annotation campaign and its main components:

1. **Segments** are text fragments that can be activated or eliminated by clicking on them.
2. **Original text** contains the initial text. Segments can be read even after being deleted.
3. **Compressed text** displays the resulting text after eliminating selected segments.
4. **Button "Restart text"** restores the initial text before removing any segment.
5. **Button "End text"** sends the compressed text to the database and displays the next text to analyze.

We have recruited 66 volunteers, all native Spanish speakers, most of them undergraduate students and in 10 weeks we have collected 2 877 summaries (48 user summaries for each text in average). The system demo can be tested on the Web[1].

6.2 Criteria for Compression

What follows sets the criteria that had to be considered by annotators. These criteria have been used to analyze each sentence individually. Moreover, it was required that the resulting compressed text had to be entirely coherent.

Conservation. At least one segment must be kept for each sentence.
Importance. The main idea of the original text must be retained.
Grammaticality. The compressed sentences should be understandable and should not have problems of coherence (e.g., sentences must have a main verb).
Brevity. It should be compressed as much as possible. This means, deleting words as long as it keeps the same meaning, but with fewer words.

All criteria are equally important.

[1] `http://dev.termwatch.es/~molina/compress4/man/`.

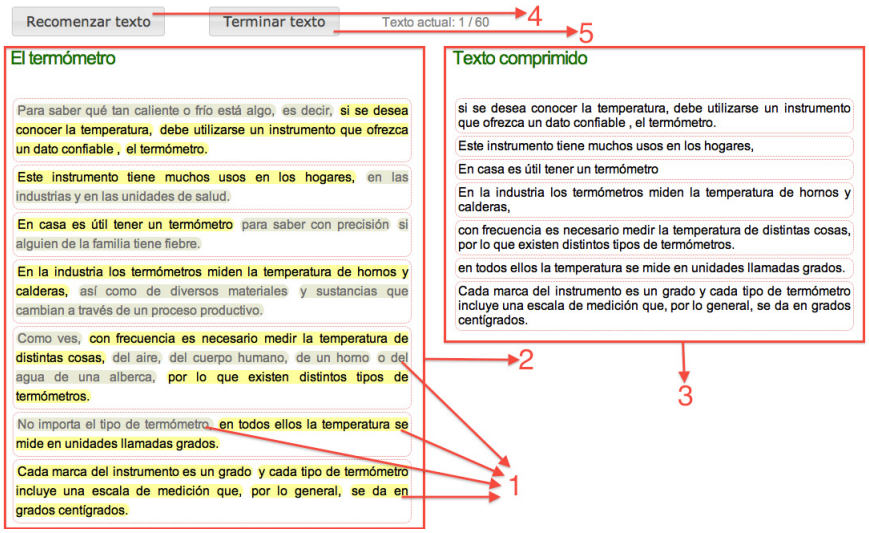

Fig. 4. Annotation system interface for the annotation campaign

7 Results

7.1 Search Space and Solution Space

The search space associated with a sentence is the number of its possible compression candidates. According to section 3.2, for a sentence with k segments, the size of its search space is $2k - 1$. In order to define which is the optimal compression of a sentence, we deal with the dilemma of deciding if one person compressed a given sentence better than another. At the moment we can only consider that if a solution was given by someone during the annotation campaign it must be considered as a candidate solution. Table 2 shows the search space size and the average solutions space size for each segmenter. In general, the search space is larger than the solution space proposed by annotators. This is a fundamental fact because it points out some regularities in the compressions proposed by the annotators.

7.2 Annotators Agreement

As seen in section 7.1, the solution space proposed is shorter than the theoretical search space. However, in most cases, there is a high degree of ambiguity in the designation of the optimal compression. As the sentence complexity increases, the variability in the number of proposed solutions also increases considerably. In many cases, the most voted compression reached only 25% of the votes. Moreover, in some cases there is no clear trend in the optimal solution. Figure 5 shows the proportion of ambiguous cases for five classes defined by the number of words

Table 2. Theoretical search space and average number of human solutions using two discourse segmenters

k	Theoretical value $(2^k) - 1$	Avg. solutions using DiSeg	Avg. solutions CoSeg
1	1	1	1
2	3	2.6	2.5
3	7	4.6	4.3
4	15	7	6.1
5	31	10.8	8.8
6	63	16	11.6
7	127	-	12.3
8	255	16	14.3
9	511	26	21
10	1023	-	18
11	2047	-	16
12	4095	-	28
13	8191	-	39
21	2097151	-	40

using different votes thresholds. In this figure only multi-segment sentences were considered (sentences with a single segment are not relevant in our analysis). In order to ensure distribution's uniformity for both segmenters, classes were defined as it follows: C1 (20 words or less), C2 (21 to 27 words), C3 (28 to 34 words), C4 (35 to 45 words) and C5 (46 words or more).

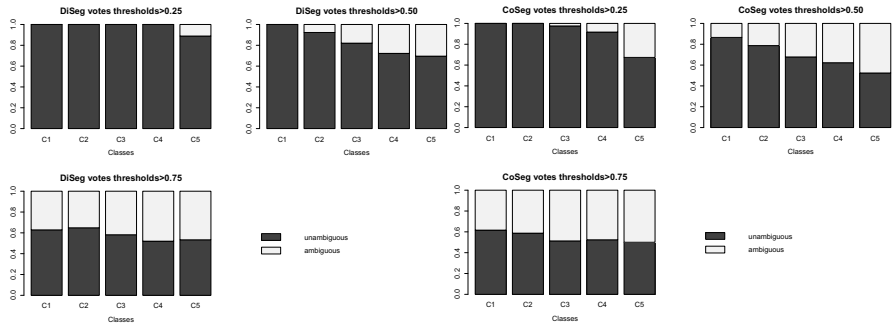

Fig. 5. Ambiguity proportion for different votes thresholds

7.3 Linear Models of Segment Removal Probability

We studied the possibility to predict the probability of whether a segment will be removed or not by annotators. We considered generalized linear models. Probability distributions are defined for each segmenter ($P_D(x)$ for DiSeg and $P_C(x)$

for CoSeg) and every segment x as the number of evaluators that chose to remove the segment over the total number of annotators. In our data, the mean probability to remove a DiSeg segment is 25% and the median probability is only 10%. These probabilities increase for the CoSeg segments reaching a mean of 32% and a median of 25%. T-test and Wilcoxon tests show that these differences are highly significant (p-value $< 10^{-}6$).

Now, if we consider only non-ambiguous deleted segments, which were removed by at least 75% of the annotators, these are 33 over 412 (7%) segments for Diseg, and 81 over 740 (10%) for CoSeg. This shows that CoSeg gives the reader more opportunity to delete single segments for compression than DiSeg. However, can the probability of deletion be estimated using segment and sentence properties? For that, we consider the following indicators:

segener: Textual energy of the segment.
sentener: Textual energy of the sentence.
eratio: segener/sentener ratio.
segw: Segment length in number of words.
sentw: Sentence length in number of words.
wratio: segw/sentw ratio.
sentlp: Sentence likelihood probability.
seglp: Segment likelihood probability.
lpratio: seglp/sentlp ratio.
segpos: Segment position in the sentence.
relpos: Segment position relative to the number of segments.
nsegs: Number of segments in the sentence.

Table 3. Linear approximation of probability distribution of removing a segment. Signif. codes: *** < 0.001, ** < 0.01, * < 0.05, . < 0.1

	DiSeg			CoSeg		
	Std. Error	t	Pr(>|t|)	Std. Error	t	Pr(>|t|)
(Intercept)	0.0607	0.91	0.3646	0.0476	6.03	0.0000 ***
segener	0.0001	-0.76	0.4490	0.0001	-2.01	0.0451 *
sentener	0.0000	0.02	0.9859	0.0000	0.29	0.7735
eratio	0.0558	-3.47	0.0006 ***	0.0714	-4.86	0.0000 ***
sentw	0.0012	-1.04	0.2977	0.0009	2.74	0.0063 **
segw	0.0021	2.31	0.0214 *	0.0029	-0.20	0.8410
sentlp	0.0000	0.24	0.8083	24902	0.63	0.5297
seglp	6.3080	-0.87	0.3841	0.9858	-1.78	0.0759
lpratio	0.0942	-6.80	0.0000 ***	0.1049	-3.94	0.0001 ***
segpos	0.0224	-1.59	0.1130	0.0083	0.11	0.9158
nsegs	0.0188	1.65	0.0998 .	0.0075	-3.05	0.0024 **
relpos	0.0819	8.51	0.0000 ***	0.0604	5.70	0.0000 ***

Table 3 shows the output of linear model fitting function, in R software, to predict segment deletion probabilities for DiSeg and CoSeg. It reveals that

deletions of CoSeg segments are correlated to a larger subset of segment and sentence descriptors. Deducing the two linear models by restricting previous linear fittings to significantly correlated indicators gives these models:

$$P_D(x) \sim 0.581\mathbf{segpos} - 0.523\mathbf{lpratio} - 0.214\mathbf{eratio} + 0.002\mathbf{segw}$$

$$P_C(x) \sim -0.342\mathbf{eratio} + 0.003\mathbf{sentw} - 1.574\mathbf{seglp} - 0.416\mathbf{lpratio} \\ -0.022\mathbf{nsegs} + 0.35\mathbf{relpos}$$

Both models are significantly correlated to the targeted probability distributions (Pearson's product-moment correlation p-value $<$ 2.2e-16). Pearson's estimate is above 0.73 for $P_D(x)$ and only above 0.57 for $P_C(x)$. This would mean that DiSeg points out segments that are easier to characterize in terms of compression based on textual energy and likelihood than CoSeg.

7.4 Evaluation

A 20% folded cross-check experiment using $\frac{7}{8}$ fraction of the corpora to predict the probability distribution over the remaining $\frac{1}{8}$ fraction of segments shows that the above linear model is an efficient inference model for $P_D(x)$ but not for $P_C(x)$. Indeed, on the 20% test sets, Pearson's estimate is above 0.69% and below 76% with a median of 0.73 for Diseg p_S meanwhile Pearson's estimate can be very low on some 20% test sets (below 10%) with a maximum of 72%.

To evaluate qualitatively the results we have designed a Turing like test using 39 abstracts. Two summaries were presented to each judge (other than the annotators): one prepared by (A) a human and the other developed by (B) the computer. The judge had to decide which of the abstracts had been accomplished by (A) and which one by (B). The final result was that the judges properly allocated 13 of the abstracts; 15 of the abstracts prepared by the computer were mistakenly taken as summaries made by humans and 14 of the abstracts prepared by humans were mistakenly taken by the computer. In brief, $\frac{2}{3}$ of the judges were confused in this game of imitation.

8 Conclusions and Future Work

In this article we have described a new method for automatic summarization by compression of discourse segments into each sentence, and using the textual energy to weight the informativeness of these segments. Thanks to our annotation campaign, we tested various interesting aspects regarding the elimination of discourse segments for the automatic summarization. Our study revealed that, in general, there is disagreement to determine the optimum compression and the degree of disagreement increases as the sentence complexity increases. However, there is a general agreement to preserve segments with high energy values. We have proposed a generalized linear model to predict the probability of deleting a

segment based on simple features. Another interesting result is that there is a human tolerance to non grammatical compressions if they allow to keep pertinent information in a short dense summary, which has led us to consider cooperative human-machine systems as an alternative to fully automatic summarization. Finally, we have performed a Turing test using the imitation game for evaluation. We believe that this kind of evaluation is more convincing than other automatic methods based on frequencies of n-grams because it implies human judging.

Acknowledgments. We would like to thank our contributors for their help with the corpus annotation. This work was partially supported by a CONACyT grant 211963 and Imagweb project.

References

1. Edmundson, H.P.: New Methods in Automatic Extraction. Journal of the Association for Computing Machinery 16, 264–285 (1969)
2. American National Standards Institute Inc.: American National Standard for Writing Abstracts. Technical Report ANSI Z39.14 – 1979, American National Standards Institute, New York (1979)
3. Witbrock, M.J., Mittal, V.O.: Ultra-Summarization: A Statistical Approach to Generating Highly Condensed Non-Extractive Summaries. In: Proceedings of the 22nd Conference SIGIR 1999, Berkeley, CA, Etats-Unis, pp. 315–316. ACM (1999)
4. Knight, K., Marcu, D.: Statistics-based summarization – step one: Sentence compression. In: Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence, Austin, TX, Etats-Unis, pp. 703–710 (2000)
5. Lin, C.Y.: Improving summarization performance by sentence compression-a pilot study. In: Proceedings of the 6th International Workshop on Information Retrieval with Asian Languages, Sapporo, Japon, pp. 1–8 (2003)
6. Hori, C., Furui, S.: Speech summarization: an approach through word extraction and a method for evaluation. IEICE Transactions on Information and Systems (Institute of Electronics, Informatics and Communication Engineering) 87, 15–25 (2004)
7. Clarke, J., Lapata, M.: Modelling compression with discourse constraints. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 1–11 (2007)
8. Steinberger, J., Jezek, K.: Sentence compression for the lsa-based summarizer. In: Proceedings of the 7th International Conference on Information Systems Implementation and Modelling, pp. 141–148 (2006)
9. Steinberger, J., Tesar, R.: Knowledge-poor multilingual sentence compression. In: 7th Conference on Language Engineering (SOLE 2007), Cairo, Egypt, pp. 369–379 (2007)
10. Sporleder, C., Lapata, M.: Discourse chunking and its application to sentence compression. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 257–264 (2005)

11. Molina, A., Torres-Moreno, J.-M., SanJuan, E., da Cunha, I., Sierra, G., Velázquez-Morales, P.: Discourse segmentation for sentence compression. In: Batyrshin, I., Sidorov, G. (eds.) MICAI 2011, Part I. LNCS (LNAI), vol. 7094, pp. 316–327. Springer, Heidelberg (2011)
12. Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: HLT-NAACL (2003)
13. Tofiloski, M., Brooke, J., Taboada, M.: A syntactic and lexical-based discourse segmenter. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. ACLShort 2009, Stroudsburg, PA, USA, pp. 77–80. Association for Computational Linguistics (2009)
14. Maziero, E., Pardo, T., Nunes, M.: Identificaç ão automática de segmentos discursivos: o uso do parser palavras. Série de relatórios do núcleo interinstitucional de lingüística computacional, Universidade de Sao Paulo, São Carlos, Brésil (2007)
15. da Cunha, I., SanJuan, E., Torres-Moreno, J.-M., Lloberes, M., Castellón, I.: Discourse segmentation for spanish based on shallow parsing. In: Sidorov, G., Hernández Aguirre, A., Reyes García, C.A. (eds.) MICAI 2010, Part I. LNCS, vol. 6437, pp. 13–23. Springer, Heidelberg (2010)
16. Afantenos, S.D., Denis, P., Muller, P., Danlos, L.: Learning recursive segments for discourse parsing. CoRR abs/1003.5372 (2010)
17. Mann, W.C., Thompson, S.A.: Rhetorical Structure Theory: A Theory of Text Organization. University of Southern California, Information Sciences Institute, California, Marina del Rey (1987)
18. Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., Padró, M.: FreeLing 1.3: Syntactic and semantic services in an open-source NLP library. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), pp. 48–55 (2006)
19. Torres-Moreno, J.M.: Résumé automatique de documents: une approche statistique. Hermés-Lavoisier, Paris (2011)
20. da Cunha, I., Fernández, S., Velázquez Morales, P., Vivaldi, J., SanJuan, E., Torres-Moreno, J.-M.: A new hybrid summarizer based on vector space model, statistical physics and linguistics. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 872–882. Springer, Heidelberg (2007)
21. Sierra, G., Torres-Moreno, J.M., Molina, A.: Regroupement sémantique de définitions en espagnol. In: Proceedings of Evaluation des Méthodes d'extraction de Connaissances Dans les Données (EGC/EvalECD 2010), Hammamet, Tunisie, pp. 41–50 (2010)
22. Fernández, S., SanJuan, E., Torres-Moreno, J.-M.: Textual energy of associative memories: performant applications of enertex algorithm in text summarization and topic segmentation. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 861–871. Springer, Heidelberg (2007)
23. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
24. Chen, S., Goodman, J.: An empirical study of smoothing techniques for language modeling. Computer Speech & Language 13, 359–393 (1999)
25. Stolcke, A.: Srilm – an extensible language modeling toolkit. In: Intl. Conf. on Spoken Language Processing, Denver, vol. 2, pp. 901–904 (2002)