Priberam Compressive Summarization Corpus:
A New Multi-Document Summarization Corpus for European Portuguese

Miguel B. Almeida[1,2], **Mariana S. C. Almeida**[1,2], **André F. T. Martins**[1,2],
Helena Figueira[1], **Pedro Mendes**[1], **Cláudia Pinto**[1]

[1]Priberam Labs, Lisbon, Portugal
[2]Instituto de Telecomunicações, Lisbon, Portugal

Abstract

In this paper, we introduce the Priberam Compressive Summarization Corpus, a new multi-document summarization corpus for European Portuguese. The corpus follows the format of the summarization corpora for English in recent DUC and TAC conferences. It contains 80 manually chosen topics referring to events occurred between 2010 and 2013. Each topic contains 10 news stories from major Portuguese newspapers, radio and TV stations, along with two human generated summaries up to 100 words. Apart from the language, one important difference from the DUC/TAC setup is that the human summaries in our corpus are *compressive*: the annotators performed only sentence and word deletion operations, as opposed to generating summaries from scratch. We use this corpus to train and evaluate learning-based extractive and compressive summarization systems, providing an empirical comparison between these two approaches. The corpus is made freely available in order to facilitate research on automatic summarization.

Keywords: European Portuguese, multi-document summarization, sentence compression, compressive summarization

1. Introduction

The automatic summarization of newswire text is a crucial problem which lies at the intersection of information retrieval and natural language processing (Luhn, 1958; Baxendale, 1958; Edmundson, 1969). The overwhelming pace of content creation on the Web makes it prohibitive for journalists and news providers to write summaries by hand; this sets a demand for automatic summarization systems that are capable to scale to large amounts of data.

We consider the problem of **multi-document summarization**, where the goal is to summarize a set of documents about the same event. Most existing systems are **extractive**, *i.e.*, they produce a summary by extracting a representative set of sentences from the original documents (Kupiec et al., 1995; Carbonell and Goldstein, 1998; Radev et al., 2000; Gillick et al., 2008). This approach has important computational advantages over abstractive systems, resulting in a simple and well-formulated optimization problem which sidesteps the need for (non-trivial) natural language generation. The resulting summaries typically have high linguistic quality, since the extracted sentences are always "grammatical" if we assume the original documents are grammatical themselves. On the other hand, extractive systems have important drawbacks when it comes to the amount of information content they can retrieve. They are quite inefficient regarding their ability to compress information, since they are obliged to leave the extracted sentences untouched. When faced with a long and partly relevant sentence, they are forced to either include it as a whole or completely discard it. This is unsatisfying, since one should be able to select the relevant parts and discard the irrelevant ones.

The drawbacks above have motivated research in **compressive summarization** (Lin, 2003; Zajic et al., 2006; Daumé, 2006; Martins and Smith, 2009), where sentences appear in the summary in a compressed form, allowing deletion of words (Knight and Marcu, 2000). Compressive systems are half-way between the extractive and the fully abstractive ones—they are still limited on the kind of summaries they can produce, but not as much as extractive systems. The problem of generating compressive summaries is more involved than in the extractive case, since it implies operating at *word* rather than sentence level. On the other hand, it allows a more fine-grained control over the content of the summary, which reflects in an increase of informativeness, as shown in recent works (Berg-Kirkpatrick et al., 2011; Woodsend and Lapata, 2012; Almeida and Martins, 2013). Most research in this field has focused on English, where good quality corpora are available. The best known are the ones associated with conference shared tasks, such as the summarization tasks of the DUC and TAC conferences,[1] which provide single- and multi-document abstractive summaries written by humans. In their work in compressive summarization, Berg-Kirkpatrick et al. (2011) used Mechanical Turk to create human-generated compressed summaries for the 48 topics in the TAC 2009 summarization dataset, using those summaries to train a compressive summarizer. Recently, the research community has begun the development of summarization corpora in languages other than English; the most prominent effort is the Multiling corpus (Giannakopoulos et al., 2011) which includes over ten languages, but not Portuguese. The only publicly available summarization corpus for Portuguese that we are aware of is CSTNews, a corpus recently introduced by Cardoso et al. (Cardoso et al., 2011), which contains 140 documents in Brazilian Portuguese (grouped in 50 topics).

In this paper, we introduce the **Priberam Compressive Summarization Corpus** (PCSC), a new corpus for multi-document summarization in European Portuguese. The corpus is made freely available in order to facilitate re-

---

[1]See `http://www.nist.gov/tac/tracks/index.html`, and links therein, for more information on the DUC and TAC summarization tasks.

search on automatic summarization.[2] PCSC covers 80 topics for events that took place in years 2010–2013, described in a list of 8 sources including newspapers, radio and TV websites. Our corpus follows a philosophy similar to Berg-Kirkpatrick et al. (2011) regarding the manual writing of the compressed summaries. Compared with the CSTNews corpus (Cardoso et al., 2011), PCSC is substantially larger (800 *vs* 140 documents), it has more sources (8 *vs* 5), it is in European rather than Brazilian Portuguese, and its manual summaries are compressive (which is useful for the learning-based approach we present here).

The following sections describe the steps taken to generate the corpus, as well as the results obtained by extractive and compressive summarization systems trained and evaluated on it.

2.   Overall Corpus Structure

The corpus consists of 80 topics, which we divided into two partitions:

- 40 topics for events that took place in 2010–2011,

- 40 topics for events that took place in 2012–2013.

In this paper, we treat the 2010–2011 section as a **training partition**, and the 2012–2013 section as a **test partition**. Note that the second partition is guaranteed to contain events which occurred only *after* those in the training set. The selection of topics involves natural disasters, sport events, political scandals (concerning international and Portuguese figures), the economic crisis, among others.

Each topic contains 10 documents written in European Portuguese, except one topic which contains 11, giving a total of 801 documents.[3] Documents were extracted from major Portuguese news sources, including:

- generalist newspapers—*Diário de Notícias* (DN), *Jornal de Notícias* (JN), and *Correio da Manhã* (CM);

- radio and TV websites with text articles—*Rádio e Televisão de Portugal* (RTP) and *TSF Rádio Notícias* (TSF);

- economy/finance newspapers—*Jornal de Negócios* (JNeg);

- sports newspapers—*Record* and *O Jogo*.

In all cases, the documents in each topic come from multiple sources, and they may cover the same event at different points in time.[4]

---

[3]The topic with the extra document is in the training portion.

[4]For example, one story may describe that an earthquake just occurred somewhere in the world, and another one could be from a few days later, with a better estimate of the number of casualties and damages.

Type	Source	Docs per source	Docs per type
Generalist	DN	129	458
	JN	168	
	CM	161	
Economy/Finance	JNeg	83	83
Sports	Record	20	23
	O Jogo	3	
Radio/TV	RTP	97	237
	TSF	140	

Table 1: Number of documents per source and per source type.

3.   Summary Annotation

To obtain the multi-document summaries for each topic, we applied the semi-automatic process we next describe. We first ran the simple (non-learned) coverage-based extractive summarizer of Gillick et al. (2008) (described summarily in section 4.) with a large upper bound of 1000 words (ten times as many as the final summary length). The goal of this step was to filter out irrelevant sentences that will not have a chance of being used in the final summary. This made the annotators' job substantially easier, and therefore faster, by pruning out sentences that are automatically determined not to be relevant. Then, human annotators were given the task of taking the sentences that survived this filtering step (about 50 on average) and producing compressed summaries with a maximum of 100 words.[5] A total of five human annotators participated in this process, with each topic being summarized by two of them.

In order to ensure that the corpus is appropriate for compressive summarization, the annotators were constrained by the following guidelines:

- Deleting whole sentences is allowed.

- Deleting parts of sentences is allowed.

- If a sentence is partially deleted, it must result in one grammatical sentence in the final summary.

- Merging partially deleted sentences into one new sentence is not allowed.

This procedure ensures that the human summaries are *attainable* by compressive summarizers, an important property of our corpus. This property ensures that one can skip the intermediate step of automatically estimating a compressive summary which approximates the human summary (normally called the "oracle"), a step that was necessary for compressive systems thus far (Martins and Smith, 2009; Almeida and Martins, 2013). By generating human compressive summaries we create a corpus where the human summaries can be attained by the automatic systems, eliminating this issue.

---

[5]The limit of 100 words was chosen to be the same as the word limits in the TAC summarization tasks.

4. Automatic Compressive Summarization

We now briefly describe the two summarizers that we evaluated on this corpus. The first of the two systems, which we call BASICEXTRACTIVE, is our implementation of the simple non-learned coverage-based system in Gillick et al. (2008). While very simple, this system is a competitive baseline, having obtained the top score in the TAC 2008 summarization shared task. The second system, here called LEARNEDCOMPRESSIVE, is a learned compressive system which led to state-of-the-art results for the same TAC 2008 dataset (Almeida and Martins, 2013).[6]

Before explaining the rationale behind these systems, we start by highlighting some simple properties any summary should have, and how each of these properties will give raise to a **component** which is tractable to optimize separately. We then describe how all components can be placed together in a factor graph and be optimized jointly using a dual decomposition approach.

4.1. Summarization Trade-Offs

Intuitively, as argued in Almeida and Martins (2013), a good summary should possess three properties:

1. **Conciseness:** It should be significantly shorter than the original document(s);

2. **Informativeness:** It should convey as much information as possible from the original document(s);

3. **Grammaticality:** It should be grammatically correct.

Hence, the process of generating an automatic summary involves a trade-off among these three qualities.

Conciseness. Following the methodologies employed at earlier DUC and TAC conferences, the conciseness property is cast simply as a hard constraint: the number of words in the summary cannot exceed a given budget B. Throughout this paper, we use the popular choice $B = 100$, consistent with the datasets from those conferences.

Informativeness. Informativeness is measured as a sum of scores for the **concepts** that are covered by the summary. Following Gillick et al. (2008), we choose as concepts all word bigrams which are not a pair of stopwords.[7] The score of each concept is non-zero only if the concept is present in the summary, regardless of its frequency: word bigrams which appear multiple times in the summary only contribute once to the score.

The actual score functions are different in the BASICEXTRACTIVE and the LEARNEDCOMPRESSIVE systems. In the BASICEXTRACTIVE system, the score of each concept is, for the topic being summarized, the number of documents (out of the 10 documents in this topic) in which the concept appears. In the LEARNEDCOMPRESSIVE system, the concepts are the same as above, but their score is given by a feature-based linear model whose weights are learned

from the training data.[8] The features used to score each concept are:

- The number of documents in which the concept appears (as in the BASICEXTRACTIVE system).

- Which of the words in the concept bigram are stopwords (only the first, only the second, or none – recall that concepts cannot have both words as stopwords).

- The earliest sentence position in which the concept appears in the topic's documents. For example, if a concept appears in the second sentence of one document, in the fourth sentence of another document, and does not appear in the other documents, this concept's earliest position is 2.

- Conjunctions of the previous features.

- A bias feature, active for all concepts.

All features are binarized; for features that are counts, we use bins to convert them to binary features (see Almeida and Martins (2013) for details).

Grammaticality. Grammaticality is not scored in the BASICEXTRACTIVE system: since we extract whole sentences, the grammaticality of the resulting summary should be ensured by the grammaticality of the sentences in the original data. In the LEARNEDCOMPRESSIVE system, this score is based on a dependency parse tree representation.

The procedure is as follows: we first tag and parse each sentence of the original document. We obtained automatic part-of-speech tags and dependency parses using Turbo-Tagger and TurboParser (Martins et al., 2013).[9] Then, we allow each node in the parse tree to be deleted, but with the following constraint: if a node is deleted, then all nodes in the subtree rooted at that node must also be deleted. In other words, our system can only delete **entire subtrees** from the parse tree.[10]

To decide which subtrees should be deleted, we compute scores for each arc in the parse tree for the three allowed possibilities: head and modifier are both included, both excluded, or the head is included and the modifier is excluded (the fourth possibility, excluding the head and including the modifier, violates the constraint mentioned in the last paragraph). These scores are again obtained from a feature-based linear model, as outlined below. The features used for compression are:

- The POS tag of the head.

- The POS tag of the modifier.

- The dependency label of the arc from the head to the modifier.

---

[6]These systems were named ICSI-1 and SINGLE-TASK, respectively, in Almeida and Martins (2013); see this reference for more detailed information on these systems.

[7]We use the list of stopwords in http://snowball.tartarus.org/algorithms/portuguese/stop.txt

[8]We also experimented with a LEARNEDEXTRACTIVE system, but since it did not outperform the BASICEXTRACTIVE one, we omit it from this paper.

[9]Both tools are available as free software at http://www.ark.cs.cmu.edu/TurboParser. The training of the tagger and parser was done using the Cintil corpus (Barreto et al., 2006).

[10]The same idea has been used by Berg-Kirkpatrick et al. (2011), but with phrase-structure trees instead of dependencies.

- The dependency label of the arc from the head's parent to the head.

- Several conjunctions of the previous features.

- Whether the arc is from a verb to a function word of that verb. For example, the verb *deixar* by itself means "to leave" or "to allow", but the expression *deixar de* means "to stop." In this case, the feature would be active in the dependency arc *deixar → de*.

- Whether the subtree being deleted contains any negation word.

- Whether the subtree being deleted is a temporal noun phrase, such as *esta terça-feira* ("this Tuesday"), or a temporal prepositional phrase, such as *até à passada terça-feira* ("until last Tuesday").

- A bias feature, active for all arcs.

All features are binary features, *i.e.*, they can be active or inactive. In addition, some arcs were forbidden from being deleted, in the sense that either both head and modifier are included, or they must both be excluded. These include arcs with the following dependency labels: SUB (verb subjects), OBJ (verb objects), VC (parts of compound verbs), PMOD (preposition modifiers), PRD (used for more than one type of predicates, most of which past participles) and DEP (a label used for punctuation modifiers).

4.2. Inference with Dual Decomposition

Unlike earlier approaches which cast compressive summarization as an integer linear program (Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011), we recently framed it as an optimization problem on a factor graph (Almeida and Martins, 2013), opening the door for efficient approximate decoding strategies. In this graph, there is one node for each word in the original document(s), representing a binary-valued variable (indicating whether the word is included or excluded from the summary). In addition, we add extra nodes for **concepts** (which, as stated in §4.1., are simple word bigrams). A concept token node is active if both of its words are included in the summary (*i.e.*, it is an AND of these two words), and a concept type node is active if any corresponding concept token is active (*i.e.*, it is an OR of all corresponding concept tokens). All word nodes are connected to a budget factor which enforces the hard constraint that the summary must not exceed B words.

Figure 1 shows an example of such a graph, for English documents. Only two sentences and one concept are shown, for clarity. Each blue box in the figure represents one sentence. All nodes inside the blue boxes represent words in those sentences. Both sentences contain the concept "Kashmiri separatists". In the top-left sentence, both words from this concept are included, as denoted by the shaded interior of the nodes labelled "Kashmiri" and "separatists". Thus the corresponding concept token is active (also shown shaded). In the bottom-right sentence, the word "Kashmiri" is excluded from the summary, as denoted by the white interior of that node. Therefore the corresponding concept token is inactive (also shown with a white interior). Since there is at least one concept token for "Kashmiri separatists" which is active, the concept type "Kashmiri separatists" (shown at the output of the OR gate) will be active.

While this summarization problem can be tackled with integer linear programming solvers, we recently devised a much faster decoding strategy which highlights the modularity of the problem (Almeida and Martins, 2013). We employ a dual decomposition framework, a class of optimization methods that tackle the dual of combinatorial problems in a modular, extensible and parallelizable way (Komodakis et al., 2007; Rush et al., 2010). We use AD[3], a free software toolkit for running dual decomposition in a customizable factor graph (Martins et al., 2011).[11]

4.3. Learning with Stochastic Subgradient Descent

To learn the feature weights, we use the same procedure as the single-task model described in Almeida and Martins (2013). Namely, we optimize a ℓ_2-regularized structured hinge loss function (a structured support vector machine) in which a cost function is defined based on the amount of words that are incorrectly deleted (with respect to a gold compressive summary) and the fraction of concepts that are missed in the predicted summary. We run 10 epochs of a stochastic subgradient descent algorithm to optimize this objective function; we use cross-validation in a development set to select the best epoch.

5. Summarization Results

As a reference for future work, we present results for the two summarization systems above on this corpus. Since each topic has two manual summaries, and following Berg-Kirkpatrick et al. (2011), we manually selected one summary in each of the 40 pairs of summaries on the training portion as the best among the two. 30 of these compressed summaries are then used to train the LEARNEDCOMPRESSIVE summarizer. Parameter tuning for the LEARNEDCOMPRESSIVE system was performed on the remaining 10 topics of the training portion; those parameters are then used to retrain the system on the whole training part of the corpus. Table 5. reports ROUGE-2 and ROUGE-SU4 recall values (Lin, 2004), two standard metrics for automatic summarization, on the 40 topics of the test portion.[12]

In order to measure the gap between automatic summaries and human summaries for this corpus, we also measured the same ROUGE scores, using each of the human summaries as a gold reference against which the other human summary for that topic is compared.[13] ROUGE scores of the human summaries are much higher than those of both automatic systems, as shown in the last row of table 5..

---

[11] Available at `http://www.ark.cs.cmu.edu/AD3`.

[12] We used version 1.5.5 of the ROUGE package, with the following options: `-l 100 -n 2 -a -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0`.

[13] Note that the automatic systems were assessed against two gold summaries, whereas each human summary is assessed against only one other human summary, therefore these scores are not fully comparable.
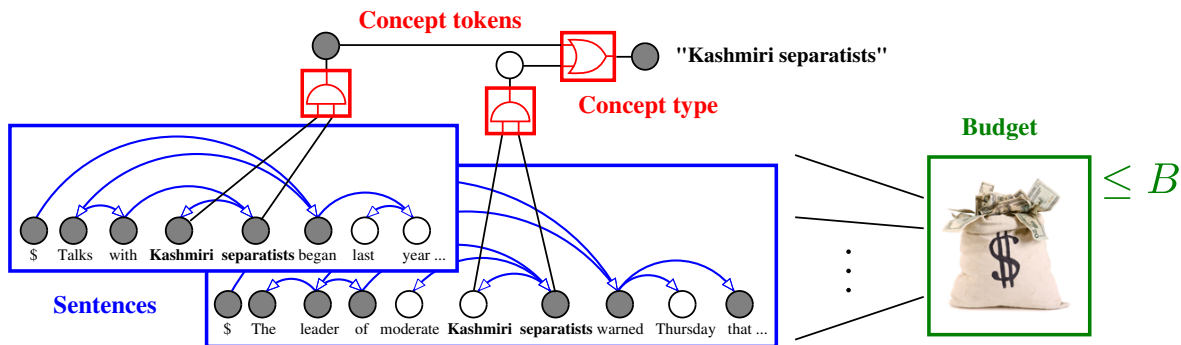
Figure 1: Components of our compressive summarizer (taken from Almeida and Martins (2013)). Each blue box represents one sentence from the original documents; nodes inside those boxes represent words which may be included or excluded from the summary. Also shown inside the blue boxes are dependency parsing arcs for these sentences, used to enforce grammaticality. The logic factors in red form the informativeness component. Finally, the budget factor, in green, is connected to the word nodes; it ensures that the summary fits the word limit. Shaded circles represent active variables while white circles represent inactive variables.

System	R-2	R-SU4
BASICEXTRACTIVE	24.54	25.22
LEARNEDCOMPRESSIVE	26.50	26.04
HUMAN	40.13	39.23

Table 2: ROUGE recall scores of the two summarizer systems on the 2012-2013 part of the corpus. Higher scores indicate better informativeness. For comparison, we also show scores obtained by human summaries.

Clearly, the gap between automatic and human summarization remains significant even when human summaries are constrained to be compressive. This demonstrates that automatic summarization systems must still be significantly improved to reach the level of informativeness of human summaries.

The ROUGE values obtained with this corpus are considerably higher than those obtained with similar systems on other corpora. For example, the LEARNEDCOMPRESSIVE system yielded 11.88 ROUGE-2 and 14.86 ROUGE-SU4 on the TAC 2008 dataset (Almeida and Martins, 2013). We speculate that this large difference could be due to our corpus having gold summaries which can be obtained as a compression of the original documents. Note that previous work exists using gold compressive summaries to *train* (Berg-Kirkpatrick et al., 2011; Almeida and Martins, 2013), but those works *test* the performance with respect to non-compressive gold summaries, which cannot be attained by compressive systems.

As an example of the kind of summaries produced by automatic systems, we present the result of the LEARNED-COMPRESSIVE system on topic 10 of the test set. Deleted words are in *gray italics*; words kept in the summary are in normal black text. Note that many other sentences were deleted completely; they are not shown for clarity.

6. Public Release

The complete corpus is available for download at the Priberam Labs website, from the URL `http://labs.`

`priberam.com/Resources/PCSC`, under a Creative Commons license.[14]
The available material includes:

- 80 topics with 10 documents in each, for a total of 800 news items in European Portuguese.

- Two manually constructed summaries for each topic, for a total of 160 summaries.

7. Conclusion

We have presented the Priberam Compressive Summarization Corpus, a new corpus for multi-document summarization in European Portuguese. It is larger than previous corpora for this language, as well as more diverse.

The corpus includes 80 topics, 10 documents per topic, and 2 manual compressed summaries per topic. This corpus allows the training of summarization systems in Portuguese, being particularly suitable for compressive systems. This corpus was used to train a state-of-the-art compressive summarizer which yielded better results than a competitive extractive system.

It is worth noting that the compressive summaries created for this corpus, as well as those produced by the learned compressive system, were constrained by a limited set of edit operations—namely, only sentence and word deletion were allowed. We demonstrated that even within these limits, humans still perform much better than automatic systems. Naturally, more complex operations (such as re-ordering words within a sentence, merging fragments from different sentences, paraphrasing certain expressions, using synonyms, among others) could be allowed to generate a richer set of summaries, and we are investigating this possibility for future work. Yet, the current gap between automatic and human performance highlights that compressive summarization remains a problem with large room for improvement, which should be tackled in order to progress to more complex approaches.

---

[14]Specifically, the license is the Creative Commons NonCommercial ShareAlike 3.0 version.

A cidade de Nova Iorque vai encerrar o metro e suspender todos os transportes públicos a partir das 19 horas deste domingo *(23 em Lisboa) devido à aproximação do furacão Sandy*, anunciou o governador do Estado. O Presidente dos Estados Unidos*, Barack Obama,* declarou hoje o estado de "*grande* catástrofe" *no estado de Nova Iorque na sequência da tempestade 'Sandy', que inundou a baixa de Manhattan e deixou meio milhão de nova-iorquinos sem electricidade. Com este novo balanço em Nova Iorque,* sobe *para 85* o número de mortos *nos Estados Unidos* afetados pelo Furacão Sandy. Quatro dias depois da passagem da tempestade pelos EUA, são agora visíveis os estragos *provocados pelos fortes ventos e chuva sentidos no país.* A tempestade 'Sandy' atingiu a costa leste dos Estados Unidos *na segunda-feira* à noite, chegando a terra a sul de Atlantic City (Nova Jérsia) com ventos de 137 quilómetros por hora.	The city of New York will shut down its subway and suspend all public transportation starting at 19 hours this Sunday *(23 hours in Lisbon) due to the approach of the Sandy hurricane*, the State governor announced. The President of the United States*, Barack Obama,* declared today the state of "*great* catastrophe" *in the state of New York in the sequence of the 'Sandy' storm, which flooded downtown Manhattan and left half a million New Yorkers without electricity. With this new tally in New York,* the number of deaths *in the United States* due to the Sandy hurricane rose *to 85.* Four days after the storm passed through the USA, the damages *caused by the strong wind and rain felt in the country* are now visible. The 'Sandy' storm hit the United States East Coast *Monday* at night, touching land south of Atlantic City (New Jersey) with 137 kilometers per hour winds.

Table 3: Example automatic summary from the LEARNEDCOMPRESSIVE system for topic 10 of the test portion of the corpus. Deleted words are in *gray italics*. The original in Portuguese is on the left; our English translation is on the right. Note that many sentences were completely deleted; they are not shown for clarity.

8. References

Almeida, M. B. and Martins, A. F. T. (2013). Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Barreto, F., Branco, A., Ferreira, E., Mendes, A., Nascimento, M. F., Nunes, F., and Silva, J. (2006). Open resources and tools for the shallow processing of portuguese: the tagshare project. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*.

Baxendale, P. B. (1958). Machine-made index for technical literature—an experiment. *IBM Journal of Research Development*, 2(4):354–361.

Berg-Kirkpatrick, T., Gillick, D., and Klein, D. (2011). Jointly learning to extract and compress. In *Proc. of Annual Meeting of the Association for Computational Linguistics*.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*.

Cardoso, P. C., Maziero, E. G., Castro Jorge, M., Seno,

E. M., Di Felippo, A., Rino, L. H., Nunes, M. d. G. V., and Pardo, T. A. (2011). Cstnews-a discourse-annotated corpus for single and multi-document summarization of news texts in brazilian portuguese. In *The Proceedings of the 3rd RST Brazilian Meeting*, pages 88–105.

Daumé, H. (2006). *Practical Structured Learning Techniques for Natural Language Processing*. Ph.D. thesis, University of Southern California.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285.

Giannakopoulos, G., El-Haj, M., Favre, B., Litvak, M., Steinberger, J., and Varma, V. (2011). Tac 2011 multiling pilot overview.

Gillick, D., Favre, B., and Hakkani-Tur, D. (2008). The icsi summarization system at tac 2008. In *Proc. of Text Understanding Conference*.

Knight, K. and Marcu, D. (2000). Statistics-based summarization—step one: Sentence compression. In *AAAI/IAAI*.

Komodakis, N., Paragios, N., and Tziritas, G. (2007). MRF optimization via dual decomposition: Message-passing revisited. In *Proc. of International Conference on Computer Vision*.

Kupiec, J., Pedersen, J., and Chen, F. (1995). A trainable document summarizer. In *SIGIR*.

Lin, C.-Y. (2003). Improving summarization performance by sentence compression-a pilot study. In *the Int. Workshop on Inf. Ret. with Asian Languages*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In Marie-Francine Moens, S. S., editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research Development*, 2(2):159–165.

Martins, A. F. T. and Smith, N. A. (2009). Summarization with a Joint Model for Sentence Extraction and Compression. In *North American Chapter of the Association for Computational Linguistics: Workshop on Integer Linear Programming for NLP*.

Martins, A. F. T., Figueiredo, M. A. T., Aguiar, P. M. Q., Smith, N. A., and Xing, E. P. (2011). An Augmented Lagrangian Approach to Constrained MAP Inference. In *Proc. of International Conference on Machine Learning*.

Martins, A. F. T., Almeida, M. B., and Smith, N. A. (2013). Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Radev, D. R., Jing, H., and Budzikowska, M. (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *the NAACL-ANLP Workshop on Automatic Summarization*.

Rush, A., Sontag, D., Collins, M., and Jaakkola, T. (2010). On dual decomposition and linear programming relaxations for natural language processing. In *Proc. of Empirical Methods for Natural Language Processing*.

Woodsend, K. and Lapata, M. (2012). Multiple aspect summarization using integer linear programming. In *Proc. of Empirical Methods in Natural Language Processing*.

Zajic, D., Dorr, B., Lin, J., and Schwartz, R. (2006). Sentence compression as a component of a multi-document summarization system. In *the ACL DUC Workshop*.