The automatic construction of large-scale corpora for summarization research

Daniel Marcu
Information Sciences Institute and Department of Computer Science
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292-6601
http://www.isi.edu/~marcu/
marcu@isi.edu

Abstract

Summarization research is notorious for its lack of adequate corpora: today, there exist only a few small collections of texts whose units have been manually annotated for textual importance. Given the cost and tediousness of the annotation process, it is very unlikely that we will ever manually annotate for textual importance sufficiently large corpora of texts. To circumvent this problem, we have developed an algorithm that constructs such corpora automatically.

Our algorithm takes as input an ⟨*Abstract*, *Text*⟩ tuple and generates the corresponding *Extract*, i.e., the set of clauses (sentences) in the *Text* that were used to write the *Abstract*. The performance of the algorithm is shown to be close to that of humans by means of an empirical experiment. The experiment also suggests extraction strategies that could improve the performance of automatic summarization systems.

1 Introduction

1.1 Motivation

All research on the automatic generation of generic abstracts assumes that the first task a summarization system needs to perform is that of *extracting* the most important units in a text. These units can be phrasal expressions [Boguraev and Kennedy, 1997], clauses [Marcu, 1999], sentences [Kupiec *et al.*, 1995, Teufel and Moens, 1997, Mani and Bloedorn, 1998, Hovy and Lin, 1999], or paragraphs [Salton *et al.*, 1994, Mitra *et al.*, 1997, Stralkowski *et al.*, 1998].

In order to train and/or evaluate the performance of an extraction engine, one cannot rely on existing ⟨*Abstract*, *Text*⟩ tuples. Although abstracts reflect in a condensed form the semantics of the texts for which they were written, most often there is no clear mapping between abstracts and the collection of important units in texts that were used to write them. For example, Teufel and Moens [1997] have found that only 31.7% of the sentences in the abstracts of the 202 articles in their corpus of computational linguistic papers could be perfectly aligned with sentences in the corresponding texts. For a corpus of 188 scientific/technical papers, Kupiec et.al. [1995] have found that only 79% of the sentences in the abstracts could be perfectly matched with sentences in the corresponding texts. And in our own work (see section 3), we have found that human judges considered that only 15% of the *clauses* in 10 abstracts taken from the Ziff-Davis corpus, a collection of newspaper articles announcing computer products, matched perfectly clauses in the corresponding texts.

To manage this linguistic gap between abstracts and texts, researchers in summarization usually employ the following methodology: A panel of judges first labels the units in a collection of texts as important or unimportant. The set of units that are considered important by a majority of judges are taken to be "gold standard". Learning-based extraction engines then use these "gold standards" in order to be trained and evaluated. Non-learning-based systems use the "gold standards" only to be evaluated. The evaluation employs traditional recall and precision figures that measure the degree of overlap between the units considered important by an extraction engine and the units considered important by a majority of the human judges.

In spite of this dependence on annotated data, summarization research is notorious for its lack of adequate corpora, a situation that prevents rapid progress in the field: today, there exist only a few small collections of texts whose units have been manually annotated for textual importance [Edmundson, 1968, Kupiec *et al.*, 1995, Teufel and Moens, 1997, Jing *et al.*, 1998, Marcu, 1999]. Given the cost and tediousness of the annotation process, it is very unlikely that we will ever manually annotate for textual importance sufficiently large corpora. To circumvent this problem, we have developed an algorithm that constructs such corpora automatically.

SIGIR '99 8/99 Berkley, CA USA

1.2 Towards the automatic derivation of corpora for summarization research

The assumption that characterizes all current summarization systems is that only certain parts of a text are important and that abstracts can be written using only these important parts, which are called *extracts*. And our own experiment (see section 3) validates this assumption: when given a collection of 10 ⟨*Abstract, Text*⟩ tuples from the Ziff-Davis corpus, 14 subjects agreed that the information presented in the abstracts reflected the semantics of only 35.88% of the texts. The semantics of the other 64.12% of the texts was never reflected in the abstract.

In the work described here, we want to automate the tedious step of manually identifying the extract, i.e., the units in a text that were used to write the abstract. More precisely, we want to take advantage of the existence of large electronic corpora of ⟨*Abstract, Text*⟩ tuples, in order to automatically derive tuples of the form ⟨*Abstract, Extract, Text*⟩, where the *Extract* component contains the clauses/sentences of the *Text* that were used to write the *Abstract*.

The potential payoff of creating large, genre-specific corpora of ⟨*Abstract, Extract, Text*⟩ tuples is two-fold:

1. First, the pairs ⟨*Extract, Text*⟩ can be used in order to train and evaluate the extraction engines of summarization systems, i.e., the engines responsible for identifying the most important parts of texts.

2. Second, the pairs ⟨*Abstract, Extract*⟩ can be used in order to train and evaluate the interpretation and generation engines of summarization systems, i.e., the engines responsible for mapping the selected textual units into coherent texts.

Determining the parts of a text that were used by the author of an abstract of that text is not trivial even for humans (see section 3). Sometimes, abstracts use clauses that can be found almost in unmodified form in the original texts. But other times, abstracts use clauses that reflect only parts of the clauses of the original text. It is possible that one clause in an abstract represents parts or all of the information given in a few clauses in the corresponding text. And it is also possible that a sentence in a text is realized as two separate sentences in the corresponding abstract. For example, when 14 subjects evaluated how the clauses in 10 abstracts from the Ziff-Davis corpus were related to the clauses in the corresponding texts (see section 3), they considered 321 times that there was a perfect match between a clause in an abstract and a clause in the corresponding text; 514 times that the meaning of a clause in a text was reflected fully by the meaning of a clause in the abstract, with the clause in the abstract realizing some additional meaning; 467 times that the meaning of a clause in the abstract was reflected fully by the meaning of a clause in the text, with the clause in the text realizing some additional meaning; 709 times that there was a certain semantic overlap between one clause in an abstract and another clause in the text; and 59 times that a clause in the abstract was not semantically related to any clause in the text, but rather, that it was written on the basis of the abstractor's knowledge, by employing an interpretation process that involved the understanding of the whole text.

Given that there is no clear match between the clauses in a text and the clauses in the corresponding abstract, *the most difficult problem that we have to solve if we are to automate the process of determining the clauses in a text that were used to write an abstract of it is that of determining the length of the extract.* Most often, there is no correlation between the length of a text, the length of its abstract, and the length of the extract that was used to write the abstract. For example, in the 10 texts in our experiments the length of the abstracts varied from 4.22% to 31.69% of the original texts, while the length of the extracts on which the human judges agreed varied from 21.46% to 56.78% of the original texts. In some cases the lengths of the abstracts and extracts were comparable (14.54% vs. 18.68% of the original text), but in other cases they were wildly different (6.49% vs. 56.78% of the original text).

2 An algorithmic approach to determining extracts

The fundamental assumption of our approach to building ⟨*Abstract, Extract, Text*⟩ tuples from ⟨*Abstract, Text*⟩ tuples is that an *Extract* corresponds to the subset of clauses in the *Text* whose semantic similarity with the *Abstract* is maximal. In the general case, given a text T of n clauses, there are $C_n^1+C_n^2+\ldots+C_n^n = 2^n-1$ extracts of non-zero length that can be built for that text. Let E_M be the extract whose similarity with the corresponding abstract is maximal. Since iterating over all possible extracts is exponential, we adopt a different approach to determining E_M.

The key idea of our approach to determining E_M is the following. Instead of answering the question "should we include this clause into the extract?", we answer a complementary question: "if we remove this clause from the text, can we still write the abstract A?". By answering the complementary question we have a clear way of determining the number of clauses that are to be included in the extract. To understand why, assume that we initially assign to E_M all clauses in the text T. Also assume that we are using a simple, normalized cosine-based similarity metric, such as that used by Hearst [1997]. If we represent both E_M and the abstract A as sequences of ⟨*t, w(t)*⟩ pairs, where t is a token and $w(t)$ is its weight, we can compute the similarity between the extract E_M and the abstract A using the formula shown in (1) below, where $w(t)_A$ and $w(t)_{E_M}$ represent the weights of token t in abstract A and extract E_M respectively.

$$sim(E_M, A) = \frac{\sum_{t \in E_M \cup A} w(t)_{E_M} w(t)_A}{\sqrt{\sum_{t \in E_M} w(t)^2_{E_M} \sum_{t \in A} w(t)^2_A}} \quad (1)$$

Input: A tuple ⟨*Abstract, Text*⟩.
Output: A tuple ⟨*Abstract, Extract, Text*⟩.

1. Break the *Abstract* and *Text* into clauses;
2. Perform stemming and delete the stop words in both sets of clauses;
   // *Build the core extract*
3. E_M = ClausesOf(*Text*);
4. **while** $((E = E_M \setminus C_i | C_i \in E_M \wedge (\forall C_j \in E_M)(i \neq j \rightarrow sim(E_M \setminus C_i, \textit{Abstract}) \geq sim(E_M \setminus C_j, \textit{Abstract}))) \wedge E > E_M)$
5. $E_M = E$;
6. **end while**
   // *Clean-up the core extract*
7. Eliminate from E_M the clauses that have a rhetorical status of *weak* satellite;
8. Eliminate from E_M the short clauses that have a rhetorical status of *strong* satellite;
9. Eliminate from E_M the subtitles;
10. Eliminate from E_M the clauses that are not similar to any clause in the Abstract;
11. Add to E_M the clauses in Text that are most similar to each clause in the Abstract;
12. Add to E_M the unique clauses in Text that contain at least two words from the Abstract that are not used in any other clauses;
13. Eliminate from E_M all short redundant clauses;
14. **return** *Extract* = E_M;

Figure 1: The extraction algorithm.

The weights of tokens are given by their frequencies in the extract and abstract respectively.

If we delete from E_M a clause C_i that is totally unrelated to the abstract A we obtain a new extract $E_M \setminus \{C_i\}$ whose similarity with A is greater than that of E_M ($sim(E_M \setminus \{C_i\}, A)$ and $sim(E_M, A)$ have equal numerators but the denominator of $sim(E_M \setminus \{C_i\}, A)$ is smaller than the denominator of $sim(E_M, A)$ because its a sum over less terms).

If we apply a greedy approach and repeatedly delete from E_M clauses so that at each step the resulting extract has maximal similarity with the abstract, we eventually converge to a state where we can no longer delete clauses without decreasing the similarity of E_M with the abstract. We consider that the extract E_M that characterizes this stage is the extract that we are looking for.

This greedy approach to determining the *Extract* component of a tuple ⟨*Abstract, Extract, Text*⟩ when the *Abstract* and *Text* components are known represents the core of our algorithm. The full algorithm is presented in figure 1. We present now its main steps.

In the first step, we use a shallow clause boundary and discourse marker identification (CB-DM-I) algorithm in order to determine the elementary textual units and the cue phrases that play a discourse role both in the *Abstract* and the *Text* [Marcu, 1997a]. The CB-DM-I algorithm explicitly encodes knowledge of the role that 450 cue phrases, i.e., phrases such as *however, in addition to* and *although*, have in signaling clause boundaries and rhetorical relations that hold between adjacent spans of text. This knowledge was derived from a corpus study [Marcu, 1997b]. The CB-DM-I algorithm, which is linear in the number of words in a text, recalls about 80% of the clause boundaries, with a precision of 90%. Once the clause-like units have been identified, their stop words are removed and the remaining words are stemmed.

The most important part of the extraction algorithm is that that builds the core extract (lines 3–6 in figure 1). The algorithm greedily determines an extract E_M of maximal similarity with the *Abstract* by deleting clauses from the original text so that the similarity of the remaining extract with the abstract is maximal at every single step. Obviously, such an approach does not guarantee that the algorithm converges towards the extract of maximal similarity with the abstract. But nevertheless, we have empirically noticed that even such a simple algorithm produces good extracts. In figure 2, we plot the similarity between the devolving extracts E_M, i.e., extracts from which clauses are deleted repetitively, and the corresponding *Abstracts* for the ten texts in our experiment (see section 3). In figure 2, a 0 abscissa corresponds to an extract that contains all n clauses in the original text, a 1 abscissa to an extract that contains $n - 1$ clauses, and so on.

As it is shown in figure 2, the similarity between a devolving extract and its corresponding abstract increases slowly as clauses are removed from the extract. The slow increase can be explained by the fact that, initially, the text contains many clauses that are not related to the abstract. By removing these clauses from the abstract, we only decrease slightly the value of the denominator in equation (1); however, the value of the numerator does not change too much. After reaching a maximum, the similarity drops sharply, the end of each graph representing the similarity between the abstract and the clause in the text that is most similar to the abstract. The sharp decrease can be explained by the fact that by the time the maximum is reached, there are few clauses left in the extract, each of them being similar to the abstract. By removing any of these clauses, it is not only the denominator of equation (1) that decreases, but also, to an even larger degree, its numerator. As figure 2 shows, the maximal similarity is reached for each text when there are only few clauses left in the extract, so presumably, the algorithm does not determine extracts that contain an unreasonable number of clauses from the original text.

Once we have built the core abstract E_M, we perform on it a set of cosmetic procedures. The first two such procedures are informed by work in the psycholinguistics of discourse

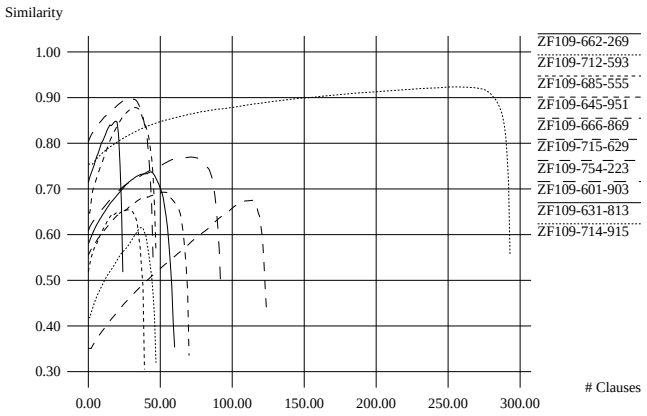

Figure 2: Behavior of the algorithm that builds the core extract.

markers and discourse relations.

Theories of discourse posit that textual units of coherent texts are related by means of rhetorical relations, i.e., relations that hold between two non-overlapping text spans called *nucleus* and *satellite*. The distinction between nuclei and satellites comes from the empirical observation that the nucleus expresses what is more essential to the writer's purpose than the satellite; and that the nucleus of a rhetorical relation is comprehensible independent of the satellite, but not vice versa.

Recent research in the psycholinguistic of discourse markers suggests that some rhetorical relations are stronger than others [Deaton and Gernsbacher, 1997, Fayol, 1997]. In order to account for this difference in strength, we have divided discourse markers into two classes. In the first class, we have put the discourse markers that signal relations whose satellites are rarely selected for summarization, i.e., satellites signalled by markers such as *instead of*, *with*, and *before*. We call such satellites *weak satellites*. In the second class we have put discourse markers that signal relations whose satellites are sometimes selected for summarization, i.e., satellites signalled by markers such as *although*, *given that*, and *since*. We call such satellites *strong satellites*. The extraction algorithm (see lines 7–8 in figure 2) removes from the extract E_M all clauses that play the rhetorical role of weak satellites and all short clauses (the clauses that have less than three non-stop words) that play the rhetorical role of strong satellites.

In our experiment, we have noticed that human judges never considered subtitles of texts to be semantically related to clauses in abstracts. To account for this, the extraction algorithm removes from E_M all subtitles.

The greedy construction of the extract E_M has employed up to this step a global measure of similarity between all clauses in E_M and all clauses in the given abstract. However, in our experiment, we have noticed that abstracts are rarely written by using single words taken from multiple clauses. It is rather that abstracts are written using sequences of words from one or many clauses. To account for this, the extraction algorithm enforces that each clause in the extract is semantically similar to at least one clause in the *Abstract* (line 10 in figure 1). The clauses in E_M whose maximal similarity with individual clauses from the *Abstract* is not above a certain threshold (0.25 in our implementation) are removed from E_M.

Since E_M is constructed in a greedy manner, it is possible that it does not reflect the global maximum of similarity with the abstract. In our experiments, we have noticed that most often it is desirable that for each clause in the *Abstract*, the extract E_M to contain the clause in the text that is most similar to it. Also, we have noticed that it is desirable to include in E_M the clauses in the text that contain at least two non-stop words from the abstract, provided that no other clauses use those words. Lines 11–12 account for adding these clauses to E_M (in the case they are not already members of it). In most cases, few text is added to E_M during the last two steps, as most of the clauses that are individually similar to those in the abstract have been already selected.

In the last step of the algorithm, we perform an internal checking of the set of selected clauses: if two clauses in E_M are very similar, the shortest is removed.

The most expensive part of the algorithm corresponds to its core, which is $O(n^2)$, where n is the number of units in the *Text*. All other steps are either linear or $O(n \times m)$, where n and m are the numbers of units in the *Text* and *Abstract* respectively. Although the algorithm is time-consuming, given the nature of the problem, this is not a major shortcoming because corpora of ⟨*Abstract*, *Extract*, *Text*⟩ tuples have to be constructed only once.

3 Experiment

3.1 Materials and method

In order to evaluate the extraction algorithm, we carried out the following experiment. We randomly selected 10 ⟨*Abstract*, *Text*⟩ tuples from the Ziff-Davis corpus, a collection of newspaper articles announcing computer products. The average length of the abstracts in the 10 tuples was 133 words and the average length of the texts was 1066 words. We ran the CBDM-I algorithm [Marcu, 1997a] and broke each abstract and each text into clause-like units. On average, the abstracts yielded 9.7 units, while the texts yielded 85.1 units. The clause-like units in each abstract and each text were labeled in increasing order with a natural number from 0 to $n-1$, where n was the number of units in the abstract and text respectively.

We presented the 10 ⟨*Abstract*, *Text*⟩ tuples of labeled elementary units to 14 independent judges and asked them to determine what were the units in the texts whose semantics was reflected by the units in the abstracts. Eleven judges analyzed all 10 tuples, while three judges analyzed only some of the tuples. Overall, we collected 125 independent judgments.

In order to determine what were the units in texts whose semantics was reflected by the units in the abstracts, we asked the judges to determine for each unit in an abstract, the units in the corresponding text that best reflected its meaning. The degree of overlap between textual units in abstracts and textual units in texts was assessed using five categories.

- If a judge considered that the meaning of a unit t in a text matched perfectly the meaning of a unit a in the corresponding abstract, they expressed that as an equality, $a = t$. For example, if $a =$ "The monochrome Sparcstation-2 40 Mhz CPU provides almost twice the performance of the Sparcstation-1 CPU." and $t =$ "The monochrome Sparcstation-2 has a 40 Mhz CPU that offers almost twice the performance of the Sparcstation-1 CPU, officials said.", then $a = t$.

- If a judge considered that the meaning of a unit t in a text was reflected entirely by the meaning of a unit a in the abstract, with a realizing some additional meaning that was not reflected by t, they expressed that as $a > t$ (a includes t). For example, if $a =$ "At $14,995, the Sparcstation-2 includes a 207Mbyte disk drive, 16Mbytes of memory expandable to 96Mbytes and up to 414Mbytes of internal disk storage." and $t =$ "The Sparcstation-2 offers as much as 414MB of internal disk storage;", then $a > t$.

- If a judge considered that the meaning of a unit a in an abstract was reflected entirely by the meaning of a unit t in the text, with t realizing some additional meaning that was not reflected by a, they expressed that as $a < t$ (a included in t). For example, if $a =$ "The 2GX costs $17,995." and $t =$ "The Sparcstation 2GX is priced at $17,995 with 16MB of memory they said.", then $a < t$.

- If a judged considered that none of the above situations applied, but nevertheless, that a and t shared some meaning, they represented the relation between the two as $a \neq t$. For example, if $a =$ "Sun Microsystem Inc. announces Sparcstation 2 and three workstations suitable for three dimensional graphics applications." and $t =$ "– Sun Microsystems last week at a press conference deployed its next generation of Sparcbased workstations, including an aggressively priced set of faster platforms aimed at 3D applications.", then $a \neq t$.

- If a judge considered that there were no units in the text that shared a significant meaning with a unit a in the abstract, they represented that as $i(a)$. Presumably, such units occurred in an abstract as a result of an interpretation process that required the understanding of parts or the whole text.

The judges were told to rely on their intuitive notion of semantic similarity in assessing the relation between units in texts and units in abstracts.

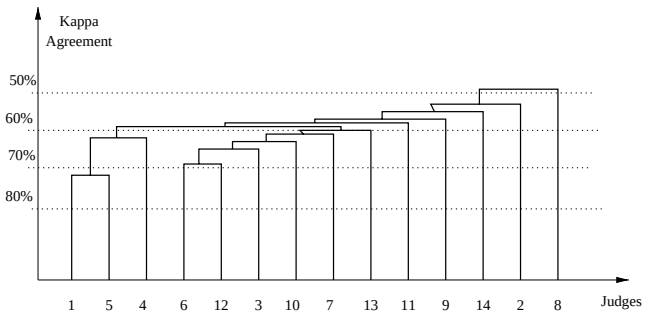

Figure 3: Bottom-up clustering of the agreement among the 14 judges — the clause level.

3.2 Results

3.2.1 Agreement

Clause-level agreement. We measured the inter-judge agreement using the kappa coefficient (K) [Siegel and Castellan, 1988], a measure that has been applied extensively in recent empirical studies of discourse. The kappa coefficient measures pairwise agreement among a set of judges who make category judgments, with correction for chance expected agreement ($K = (p(A) - p(E))/(1 - p(E))$, where p(A) is the proportion of times in which the judges agree and p(E) is the proportion of times that one would expect them to agree by chance).

In an attempt to detect outliers, we computed K using a bottom-up, hierarchical clustering procedure. By analyzing the dendogram of the clustering results in figure 3, one can notice that there are two distinct groups of judges that are characterized by agreement figures above 60%, which is the traditional threshold that is considered to reflect the existence of substantial agreement. The first group is comprised of three judges (1, 5, and 4) and the second group of six judges (6, 12, 3, 10, 7, 13). When these two groups are clustered, they yield a group characterized by a kappa coefficient of 58.85%. The other five judges who participated in the experiment monotonically decrease the kappa agreement when added to the clustering group.

The results in figure 3 show that there is substantial variation with respect to the notion of semantic similarity that human judges employ. This variation is natural given that the experiment relied completely on judges' intuitions: for example, some judges considered two textual units to match perfectly (they used equality) only when the units used exactly the same words; in contrast, other judges considered two textual units to match perfectly even when the semantic similarity arose from the use of synonyms, antonyms, or hypernyms. Nevertheless, we believe that judgments pertaining to the nine subjects whose kappa agreement was 58.85% can be safely taken as "gold standard" for semantic similarity.

The experiment described in this section provides a very fine-grained perspective on the relationship between abstracts

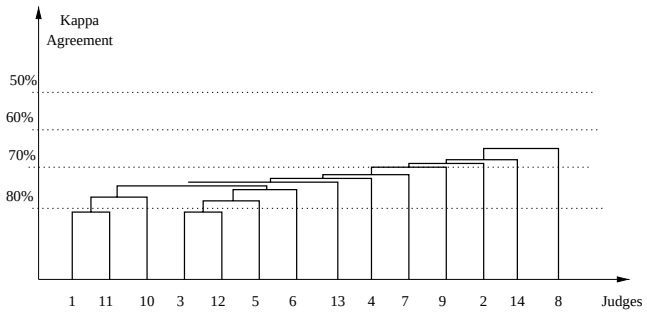

Figure 4: Bottom-up clustering of the agreement among the 14 judges, for the case of modified, binary judgments — the clause level.

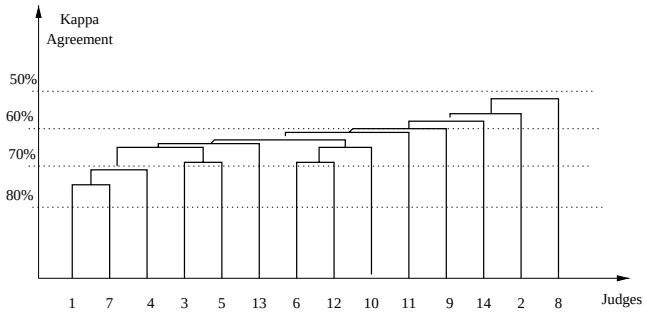

Figure 5: Bottom-up clustering of the agreement among the 14 judges — the sentence level.

and extracts, a relationship that we plan to explore in future work on generating abstracts. However, in order to evaluate the extraction algorithm that was presented in section 2, we need to consider only a watered-down version of the judgments. Since we are interested to assess whether the extraction algorithm correctly detects the units in a text that were used to write the corresponding abstract, we rewrote the similarity judgments using a binary scale. According to this scale, a unit of text was considered to be important for summarization if it was judged to be semantically similar ($=, >, <,$ or $\neq$) with at least one unit in the corresponding abstract.

When we apply the bottom-up, hierarchical clustering procedure on the binary judgments, we obtain the dendogram in figure 4. As one can see, when one assesses only whether a unit in a text is semantically related to a unit in the corresponding abstract, the agreement among judges is significant: the agreement of 11 judges is above 70% and the agreement of all 14 judges is 64.47%.

Sentence-level agreement. The majority of summarization systems use sentences as elementary units of interest. Since we are interested to automatically build adequate summarization corpora for such systems as well, we also assessed the ability of human judges to determine the sentences (not clauses) in a text that were used to write the abstracts. In order to do this, we automatically rewrote the clause-level similarity judgments, so that they reflected sentence-level similarities: if a clause of a sentence in a text was judged to be semantically related to a clause of a sentence in the corresponding abstract, we assigned the same semantic judgment to the relation between the two sentences.

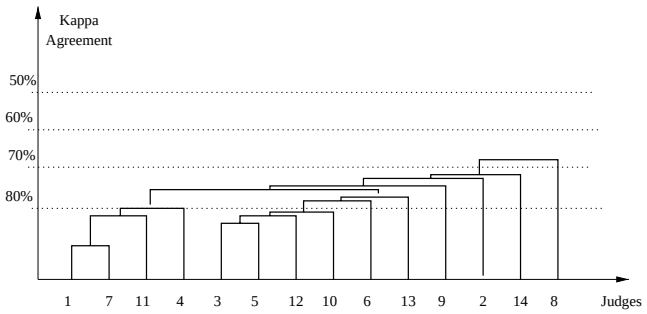

Figure 6: Bottom-up clustering of the agreement among the 14 judges, for the case of modified, binary judgments — the sentence level.

Figures 5 and 6 show the dendograms produced by the clustering procedure. At sentence level, the kappa coefficient among 11 judges is 60.62% in the case of fine-grained judgments. When the clustering procedure is applied to binary judgments, 13 judges agree at 72.55% level. Although judge 8 is an outlier, he reduces the agreement to only 68.53%, which is still statistically significant.

3.3 Evaluation of the performance of the extraction algorithm

In order to evaluate the performance of the extraction algorithm, we first used the human judgments in order to determine an upper-bound of the average human performance on the task of identifying the extracts that were used to write abstracts. For each judge, we determined the set of units from each text that were considered by a majority of the other 13 judges to be similar with units in the corresponding abstract. We hence determined for each text the "gold standard" extract on which a majority of 13 judges agreed and then compared the extract produced by the fourteenth judge with the gold standard. The comparison employed traditional recall and precision figures, which reflect the percent of textual units correctly selected by a judge with respect to the gold standard and the percent of textual units correctly selected by a judge with respected to the total number of selected units respectively.

In addition, we computed the recall and precision results that characterized the extracts constructed automatically, by the extraction algorithm. In this case, the recall and precision figures were computed with respect to gold standards that characterized the majority opinion of all 14 judges. Table 1 shows the average recall and precision results that characterize the human judgments for each document, as well as the results of the extraction algorithm. To enable a better comparison of the manual and automatic approaches, in addition

Doc	Judges			Extraction algorithm		
ZF109-	Rec	Prec	F-val	Rec	Prec	F-val
662-269	84.80	74.36	78.15	82.35	73.68	77.78
714-915	76.97	63.31	68.32	68.75	52.38	59.46
712-593	88.21	78.24	81.56	56.25	75.00	64.29
685-555	97.80	93.10	94.96	85.71	60.00	70.59
645-951	95.86	84.50	89.43	100.00	92.86	96.30
666-869	82.69	75.28	77.69	57.14	66.67	61.54
715-629	80.17	79.50	78.95	72.73	80.00	76.19
754-223	90.20	91.90	90.51	100.00	85.00	91.89
601-903	89.17	70.22	77.44	80.00	57.14	66.67
631-813	94.23	99.04	96.4	100.00	100.00	100.00
Average	88.01	80.94	83.34	80.29	74.27	76.47

Table 1: Performance of the extraction algorithm — the clause level.

Doc	Judges			Extraction algorithm		
ZF109-	Rec	Prec	F-val	Rec	Prec	F-val
662-269	86.67	86.70	85.90	80.00	85.71	82.76
714-915	78.18	68.14	71.54	66.67	57.14	61.54
712-593	90.77	78.95	83.41	63.64	87.50	73.68
685-555	97.44	92.86	94.63	83.33	62.50	71.43
645-951	97.20	86.86	91.42	100.00	91.67	95.65
666-869	80.00	80.83	79.56	54.55	66.67	60.00
715-629	82.83	79.72	80.36	77.78	77.78	77.78
754-223	91.07	92.83	91.50	92.86	81.25	86.67
601-903	88.64	77.34	81.75	81.82	64.29	72.00
631-813	94.51	100.00	97.04	100.00	100.00	100.00
Average	88.73	84.42	85.71	80.06	77.45	78.15

Table 2: Performance of the extraction algorithm — the sentence level.

to the recall and precision figures, we also show in table 1 the F-values ($F = 2 \times \mathit{Rec} \times \mathit{Prec}/(\mathit{Rec} + \mathit{Prec})$).

As table 1 shows, the performance of the extraction algorithm comes very close to the average performance of human judges, the F-value pertaining to the program being about only 7% smaller than that pertaining to the judges. The results in table 2, which depict the recall, precision, and F-value figures that pertain to sentence-level extracts, show the same difference in performance between humans and the extraction algorithm. The automatically generated extracts that correspond to the results shown in table 2 were obtained by simply extending the boundaries determined by the extraction algorithm in figure 1 from clause-level to sentence-level. That is, if a sentence s contained a clause c that was selected by the extraction algorithm, then the whole sentence s was considered to be part of the sentence-based extract.

When we tried to apply the extraction algorithm directly at sentence level, we obtained recall and precision results that were about 10% lower than the results reported in table 2. This suggests that clauses are better than sentences for the task of determining the extracts that comprise all the information in a text that is realized in an abstract of it. When we determined the extracts using only the core algorithm (lines 3–6 in figure 1), we obtained 65.11%, 65.58%, 64.7% recall/precision/f-val figures at the clause level and 68.43%, 71.54%, 69.26% at the sentence level. This shows that the core algorithm, which is general, does most of the job.

4 Discussion

4.1 Discussion of the approach

The most important criticism that can be brought up in connection with our approach to building large corpora for summarization research is that it assumes that abstracts objectively represent in a condensed form the most important parts of texts. However, as it has been often emphasized, abstracts are subjective artifacts: for a given text, it is highly likely that different people would build different abstracts. To a certain extent, when the units in a text are labeled manually for textual importance by a large number of judges, the subjective nature of the enterprise is factored out. We believe that the same argument holds at the corpus level as well: although it is possible that the abstracts of single documents are highly subjective, when one considers abstracts of thousands of documents the subjective nature of the entire corpus is factored out.

4.2 Discussion of the methodology

Obviously, the extraction algorithm that we presented in this paper is highly sensitive to the notion of semantic similarity that it employs and to the nature and size of the semantic units that it computes with. In our experiments, we have tried a variety of methods: we have run versions of the algorithms that did not delete the stop words from the clauses, that employed morphing instead of stemming, that did not employ any kind of morphing and stemming, and combinations of them. The combinations and the order of the steps shown in figure 1 produced the best results. However, many other combinations came close to the performance results discussed in section 3.3 (within approximately 5 to 10%).

Besides computing recall and precision figures, we have also manually compared the extracts determined by the program and the extracts on which the judges agreed. Our visual comparison suggested that the simple notion of semantic similarity that we employed here can provide an adequate solution to the problem of building *Extracts* from $\langle$*Abstract*, *Text*$\rangle$ tuples. The cases in which a more sophisticated measure of similarity would have been required in order to determine that two units were related did not occur frequently. For example, in one such case, the abstract talked about the *weight* of a computer, while the text just mentioned that the computer was *heavy*. By applying WordNet-based notions of semantic similarity we can presumably detect the similarity between *weight* and *heavy* [Fellbaum, 1998]; however, it is not clear yet how much improvement in performance we will get from that.

4.3 Discussion of the results

From a summarization perspective, the most interesting result pertains to the length of the extracts. The average length of the ten extracts on which the judges agreed was 367 words.

That means that in order to generate abstract that represented on average 17.59% of the length of the texts, professional abstractors needed to rely on 35.88% of the texts (at least for the text genre of the Ziff-Davis corpus). This corresponds to a 2.76 average ratio between the length of an extract and the length of the abstract that is generated from it. This result strongly suggests that in order to build abstracts, automatic summarization systems will have to first select important units that comprise about two times and a half the number of words that will be eventually used in the abstract. The simple catenation of the selected units will not suffice. Rather, it seems that an interpretation stage is required, one that would eliminate the units that are redundant and that would present the information in the selected units in a more compact form.

We have used the extraction algorithm in order to create a corpus of 6942 ⟨*Abstract, Extract, Text*⟩ tuples using as input ⟨*Abstract, Text*⟩ tuples from the Ziff-Davis corpus. The average length of the 6942 texts was 970 words and the average length of the abstracts was 123 words. The lengths of the automatically generated extracts was 283 words. The abstract to text compression rate was 17.86%, and the extract to text compression rate was 34.70%. These figures are very close to those that characterized the compression rates associated with the 10 texts for which extracts were determined manually (17.59% and 35.88% respectively). This corpus is already used by a number of summarization research groups.

References

[Boguraev and Kennedy, 1997] Branimir Boguraev and Christopher Kennedy. Salience-based content characterisation of text documents. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 2–9, Madrid, Spain, July 11 1997.

[Deaton and Gernsbacher, 1997] J.A. Deaton and M.A. Gernsbacher. Causal conjunctions and implicit causality cue mapping in sentence comprehension. *Journal of Memory and Language*, 1997.

[Edmundson, 1968] H.P. Edmundson. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285, April 1968.

[Fayol, 1997] Michel Fayol. On acquiring and using punctuation: A study of written French. In Jean Costermans and Michel Fayol, editors, *Processing Interclausal Relationships. Studies in the Production and Comprehension of Text*, pages 157–178. Lawrence Erlbaum, 1997.

[Fellbaum, 1998] Christiane Fellbaum, editor. *Wordnet: An Electronic Lexical Database*. The MIT Press, 1998.

[Hearst, 1997] Marti A. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, March 1997.

[Hovy and Lin, 1999] Eduard Hovy and Chin-Yew Lin. Automated text summarization in summarist. In Inderjeet Mani and Mark Maybury, editors, *Advances in Automatic Text Summarization*. The MIT Press, 1999. To appear.

[Jing *et al.*, 1998] Hongyan Jing, Regina Barzilay, Kathleen McKeown, and Michael Elhadad. Summarization evaluation methods: Experiments and analysis. In *Proceedings of the AAAI–98 Spring Symposium on Intelligent Text Summarization*, pages 60–68, Stanford, 1998.

[Kupiec *et al.*, 1995] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th ACM/SIGIR Annual Conference on Research and Development in Information Retrieval*, pages 68–73, Seattle, Washington, 1995.

[Mani and Bloedorn, 1998] Inderjeet Mani and Eric Bloedorn. Machine learning of generic and user-focused summarization. In *Proceedings of Fifteenth National Conference on Artificial Intelligence*, Madison, WI, 1998.

[Marcu, 1997a] Daniel Marcu. The rhetorical parsing of natural language texts. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL–97)*, pages 96–103, Madrid, Spain, July 7-12 1997.

[Marcu, 1997b] Daniel Marcu. *The rhetorical parsing, summarization, and generation of natural language texts*. PhD thesis, Department of Computer Science, University of Toronto, December 1997.

[Marcu, 1999] Daniel Marcu. Discourse trees are good indicators of importance in text. In Inderjeet Mani and Mark Maybury, editors, *Advances in Automatic Text Summarization*. The MIT Press, 1999. To appear.

[Mitra *et al.*, 1997] Mandar Mitra, Amit Singhal, and Chris Buckley. Automatic text summarization by paragraph extraction. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 39–46, Madrid, Spain, July 11 1997.

[Salton *et al.*, 1994] Gerard Salton, Chris Buckley, and Amit Singhal. Automatic analysis. Theme generation and summarization of machine-readable texts. *Science*, 264:1421–1426, June 3 1994.

[Siegel and Castellan, 1988] Sidney Siegel and N.J. Castellan. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, Second edition, 1988.

[Stralkowski *et al.*, 1998] Tomek Stralkowski, Jin Wang, and Bowden Wise. A robust practical text summarization. In *Working Notes of the AAAI–98 Spring Symposium on Intelligent Text Summarization*, pages 26–33, Stanford, CA, March 23–25 1998.

[Teufel and Moens, 1997] Simone Teufel and Marc Moens. Sentence extraction as a classification task. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 58–65, Madrid, Spain, July 11 1997.