Identifying Information Units for Multiple Document Summarization

Seamus Lyons, Dan Smith

School of Computing Sciences, University of East Anglia,
Norwich, NR4 7TJ, UK

Abstract. Multiple document summarization is becoming increasingly important as a way of reducing information overload, particularly in the context of the proliferation of similar accounts of events that are available on the Web. Removal of similar sentences often results in either partial or unwanted elimination of important information. In this paper, we present an approach to split sentences into their component clauses and use these clauses to produce comprehensive summaries of multiple documents describing particular events. Detailed analysis of all clauses and clause boundaries may be complex and computationally expensive. Our rule-based approach demonstrates that it is possible to achieve high accuracy in reasonable time.

1 Introduction

Thanks to the explosion of information on the Web, there are an increasing number of situations where we have multiple partial reports of an event that would be more easily assimilated as a single comprehensive summary. In this paper we address the problem of producing a comprehensive summary from a set of documents containing partial descriptions of an event.

Traditional language processing techniques use the sentence as the text unit to analyze, and generally assume well-formed grammatical sentences. The journalistic texts we are interested in here exhibit a number of features that complicate their processing by conventional approaches. For example, sentences are broken for emphasis, regardless of the grammatical consequences, ideas may span several short paragraphs, sentences are added together, and clauses are shortened and added to the previous sentence. Human readers typically have sufficient background knowledge of the structure of events being described, idioms, etc. to easily understand and summarize these texts. Our approach to automatically produce summaries relies on an initial phase of domain-specific information extraction, which allows us to restrict the problem to that of summarizing a set of documents that we know are all describing the same event. We have performed our initial experiments on reports of soccer matches, as they describe well-defined and structured events, but typically exhibit all the language problems we are concerned with. Many other domains have a similar proliferation of sources; for example, an analysis of Netnews articles found that 18% of articles overlap by 80% or more [1].

Lyons S. and Smith D. (2005).
Identifying Information Units for Multiple Document Summarization.
In *Proceedings of the 2nd International Workshop on Natural Language Understanding and Cognitive Science*, pages 110-117
DOI: 10.5220/0002564701100117

Soccer matches are reported from source to source. The diversity in coverage results in a large resource of informative extracts that differ but also overlap. Similar segments are grouped together and tokenized to isolate clausal units of information. The major hurdle to clause identification is the ambiguous use of conjunction terms. The main focus of this paper is to describe our approach to clause boundary detection and to present the results of our experiments using this approach. We also describe a prototype system that provides a unified version of documents from several web sources with overlapping content.

The rest of this paper is structured as follows. Related work is described in section 2. Section 3 describes the set of rules that classify clause boundary candidates and the clause-splitting algorithm. The duplicate clause detection algorithm is described in section 4. Initial experimental results are presented in section 5. Conclusions and future work are discussed in section 6.

2 Related Work

Summarization systems [2, 3] use statistical methods to determine which sentences are important when creating an extract from a text. Size reduction has traditionally been the primary concern of summarization systems [4]. This produces a series of sentences classified as the most important based on the relationship of the title and sentence words, or the length and position of the sentence in the document. These systems achieve the task of text reduction using a range of NLP and machine learning techniques as exemplified at the DUC conferences [5].

Work on Multiple Document Summarization (MDS) systems has shown the importance of sentence order [6], the coherence of the text and, the integration of text. A similarity metric based on common terms in each sentence indicates duplication and importance. Similar sentences are classified to belong to the same cluster, called a *theme*. In some MDS systems [7, 8], the application of complex linguistic analysis determines the main constituents in the sentences. These are fused together using techniques such as dependency trees [9]. Problems occur as the statistical approach results in the initial sentences gaining the highest importance metric. Even in the best systems, such as Newsblaster[1], the final summary is formed from the initial section of the majority of sources. The main criteria remains size rather than relevance so important information is lost as the size reduces. In addition, the removal of sentences does not address the content of parts of the sentence. This results in partial or unnecessary removal of important or duplicate information.

The integration of all related segments into one coherent text is the primary goal of the Information Mediation System. This has been designed not include complex NLP techniques. A hierarchical approach allows a region of a document, such as a text body, to be extracted. This is segmented in the extraction stage described in [10], followed by identification of sub-sentence information units, and their integration into a unified account. Research in clause identification and approaches to clause boundary identification are seen in the CoNLL-2001 shared task [11]. In [12, 13] the

[1] http://www1.cs.columbia.edu/nlp/newsblaster/

use of a context window contributes to the calculation of clause boundary use. A context window views the terms, and the syntactic group of the terms, before and after a selected term. Although these systems use advanced techniques we use the context window and sentence positioning as the basis of the clause splitting process.

3 Clause Identification

To split a sentence into clauses that represent a unit of information, the system must identify non-sentence boundaries. The clause splitter applies a set of rules to determine if a candidate clause marker indicates a clause boundary or not. Given the identification of all successful candidates, the sentence is categorized and split into clauses. Locating clause boundaries involves the resolution of the ambiguous use of punctuation and conjunctional terms. For example, commas can have a serial meaning, they can be incorrectly omitted or authors can overuse commas to help the reader understand the sentence. Therefore, all conjunction instances must be considered as possible boundary candidates.

The corpus (Collection 1) used in the work described here is a collection of soccer reports of English Football League matches taken from three sources, comprising approximately 35,000 words in 103 reports. The rules were derived from the analysis of 20 reports and the remaining 83 were used for testing.

Many terms are used as conjunctions, and many of them have other roles. Initial research showed that a small set of candidates could be used to locate the majority of boundary instances. These include coordinating conjunctions {and, but}, WH-clause subordinating conjunctions {when, where, which, while, who, whose}, other subordinating conjunctions {as, after, before, although}, and punctuation marks {commas, dashes, colons, semi-colons, brackets, underscores}. An analysis of Collection 1 showed that it contains 1810 instances of a clause boundary of which 1754 (96.9%) was attributed to the candidate set listed above.

The correctness of a candidate used to split a sentence is more important than the omission of a boundary as an incorrect split is likely to result in meaningless clauses. The primary concern is to compile a set of simple rules to identify the use of a candidate in another syntactic role. The correct candidate function, denoted '$_{corr}()$', identifies a clause boundary. This is determined by sentence position and the context of the candidate stated in the rule definitions seen in figure 1. The algorithm identifies clause endings, splits the preceding text as a clause, and leaves the remaining text as the last clause. Incorrect candidates are identified when meeting the conditions listed in the rule definitions and ignored in the splitting process. Following [13] we use a context window of +/- 3 terms. We apply a set of six rules to determine if a candidate is a boundary between two clauses. Rules 1, 2 and 6 are positional rules whilst rules 3, 4 and 5 involve the context window. These are enforced in this order according to the success rate in the experiments.

Let Sentence S be a string of terms[2] t of the length n where $t_e \in S$, and $1 \leq e \leq n$. $Z = \{(t_s, t_e) \mid 1 \leq s < e \leq n\}$ where t_s and t_e denotes a clause start and end terms.
$C = \{ j \mid j$ is a candidate$\}$ and $c \geq 0$ where c is the number of successful candidates.

[2] A term is defined as a word, a number or a word equivalent (e.g. a date).

	$_{corr}(t_e)$ = 1	
Rule 1	if	$e = 1$ or $e = n$
Rule 2	elsif	($e < 4$ or $n - e < 3$) and $_{type}(S) \neq NR$
Rule 3	elsif	t_e = "as" and t_{e+2} = "*as*"
Rule 4	elsif	t_e = "*and*", and $t_{e+1\ pos} = t_{e-1\ pos}$, and $t_{e-1\ pos}$ is a noun
Rule 5	elsif	$t_e \in$ {"before", "after"}, and t_{e+2} = d $\in$ D_1 = *{"break", "interval"}*, or t_{e-1} contains d $\in$ D_2 or t_{e-2} contains *d* $\in$ D_2 = *{"minute"}*
Rule 6	elsif	$t_{e+1} \in C$
	else	$_{corr}(t_e)$ = 0

Fig. 1. The set of rules to determine the correct use of a candidate as a clause boundary

Positional Rules. If the candidate is the first or last term in the sentence, it is not used to split a sentence (rule 1). If the candidate is in the first four or last three terms in the sentence, and does not possibly contain a central non-restrictive clause, it is excluded from consideration (rule 2). A sentence with a central non-restrictive clause is either a sentence with multiple commas or dashes. Rule 6 covers the instances where adjacent candidates are jointly used to signify a boundary.

Context Rules. In rule 3, the candidate is a subordinate conjunction used as part of an adverbial connective (e.g. "as far as”," as soon as"). The system uses the Brill parts-of speech tagger before the integration stage to create a lexicon of extracted terms and their syntactic group. If the candidate term is "and", and the parts of speech of the terms before and after this are both nouns, it is deemed not be a clause boundary. This occurs in lists, phrases and alternative constituent conjunction. For example, the phrases "apple and pears" and "high and low" use the term "and" in a non-boundary role.

There are phrases used in different domains that affect the use of clause candidates. The context window is used to correct this. Rule 5 shows the value of enforcing additional domain-specific rules. Each set of domain terms (D_1, D_2) can be extended and the context window size increased or reduced to suit the domain. In the soccer domain, the terms "before" and "after" are frequently used to refer to temporal comparisons. For example, a soccer match is split into two halves with the half-time period referred to as the "interval" or the "break”. The use of the condition "contains" in contrast to "equals" (seen in rule 5 in figure 1) refers to the ability to differ from an exact textual match by the use of a regular expression. This allows for the pattern match of both terms "minute" and "minutes". Although rule 4 and 6 identify a comma used with the term “and” in a list, there remains a need to check for serial commas. A function to identify possible lists exists prior to the clause-splitting algorithm. These are classified as $_{type}(S)$ ='L'. Other categories include simple (A), compound or complex (B), non-restrictive (NR), multiple claused (X) and sentences that contain a quote (Q). These categories are determined by pre- and post-algorithm conditions. The pre-conditions include: if there are no candidates (j $\notin$ S) then $_{type}$(S) = 'A', if S contains a quotation mark then $_{type}$(S) = 'Q', and if S contains two dashes then $_{type}$(S) = 'NR'. The post-conditions state if the number of successful candidates c = 0 then $_{type}$(S) = 'A', if c = 1 then $_{type}$(S) = 'B' and if c = 2 then $_{type}$(S) = 'X'.

4 Theme Integration

Text segments containing similar information are grouped together into *themes*. The segments are split into clauses and re-analyzed for similarity. The similarity measure is based on normalized term frequency and clauses are classed as similar if the similarity measure exceeds an empirically-defined threshold. Figure 2 shows the clause-splitting algorithm. If the similarity measure between two texts is above the given threshold then the texts undergo a clause resolution stage. If a text can be reduced into clauses and a clause has a higher similarity measure than the complete segments then there is a possibility that at least one text can be reduced.

Let Sentence S be a string of terms[3] t of the length n where $t_e \in S$, and $1 \leq e \leq n$. $Z = \{(t_s, t_e) \mid 1 \leq s < e \leq n\}$ where t_s and t_e denotes a clause start and end terms.
$C = \{ j \mid j$ is a candidate$\}$ and $c \geq 0$ where c is the number of successful candidates.

```
s = 1, e = 1, c = 1
while e ≤ n
  if t_e ∈ C and corr(t_e) = 1
  then Z_c = {t_s, t_{s+1},…,t_{e-1}}, c = c + 1, s = e
  e = e + 1
end while loop
Z_c = {t_s, t_{s+1},…, t_e}
```

Fig. 2. The clause-splitting algorithm

Figure 3 shows two segments extracted form the text corpus. The segment *X* is a compound sentence. Two independent clauses are joined by a coordinating conjunction ("and") preceded by a comma. The segment *Y* contains a subordinating conjunction. In this segment, the second clause (*y2*) is dependent on the first (*y1*). These sentences could be re-analyzed as coordinating and subordinating clauses but they both display the complexity of web text characterized by the occurrence of restrictive and non-restrictive clauses. The second clause in segment *X* (*x2*) is independent but begins with a pronoun. This refers to the referent "Gillingham" contained in the first clause (*x1*). This relationship is covered by a separate function that deals with co-reference by giving each text an address and noting the connection of these addresses. This linking process is taken in consideration when the text is removed. Although sentence extraction systems have analyzed the co-reference problem with considerable success [14], these are not implemented here.

A restrictive clause is a crucial element of a sentence. The deletion of the clause would change the meaning or render the sentence incomprehensible. Alternatively, non-restrictive clauses do not modify the preceding clause subject and deletion can occur without impairing the meaning of the remaining sentence. Non-restrictive clauses appear in the middle and at the end of sentences [15]. Therefore, if a system can identify a non-restrictive clause containing no relevant information, it can remove it.

Segment *X* is split into three clauses *x2*, *x3* and *x4* respectively. The initial independent clause (*x1)* consists of an independent clause (*x3)* and a non-restrictive clause (*x4)*. The non-restrictive clause (*x4)* is dependent on the independent clause

[3] A term is defined as a word, a number or a word equivalent (e.g. a date).

(*x3)*. If this clause (*x4)* is analyzed as not similar to any clauses in segment *Y*, it can be removed without adversely affecting the sentence coherence.

[$_X$ [$_{x1}$ [$_{x3}$ **Gillingham** had to play **without player-manager Andy Hessenthaler**],
[$_{x4}$ sidelined with **a serious knee injury**]], and
[$_{x2}$ they also hoped to shrug off the other off-field distractions caused by newspaper reports linking the club with allegations of financial irregularities.]]
[$_Y$ [$_{y1}$ **Gillingham** were **without player-manager Andy Hessenthaler**],
[$_{y2}$ [$_{y3}$ on crutches],
[$_{y4}$ following a **serious knee injury** sustained against Bournemouth in the FA Cup the previous week.]]]

Fig. 3. Segments after the clause splitting process (common features in bold).

Segment *Y* is a sentence with a non-restrictive clause separating a subordinate clause from the main clause (*y1)*. The non-restrictive clause, (*y3)* in figure 3, can be removed and the surrounding text re-joined to form a normal subordinating clause (*y1+ y4)*. This occurs when it is not deemed similar to any clauses in segment *X*. The clauses are re-analyzed to determine if it is necessary to split the segments to gain greater similarity. The similarity between segment *X* and segment *Y*, *sim* (*X*, *Y*), is 0.09738. This is above the threshold of 0.05 used in this analysis.

The clause resolver initially joins the clauses that have a similarity measure above the similarity threshold. In the example given, *clause 1*($_{x3}$) and *clause 2*($_{x4}$) in segment *X* are considered similar to *clause 1*($_{y1}$) and *clause3* ($_{y4}$) in segment *Y*. If they pose no co-reference resolution problems and no size requirements, preference is given to adjacent clauses, as there are no coherency problems. If there is no difference between segments, then preference is given to the first (segment *X*).

[$_S$ [$_{s1}$ **Gillingham** were **without player-manager Andy Hessenthaler**],
[$_{s2}$ sidelined with **a serious knee injury.**]]

Fig. 4. The two text segments are integrated into one version.

Figure 4 displays the integrated version. The system allows for size requirements considerations. If the complete version requires a reduced text then the theme shown in figure 4 has the secondary clause (*s2)* deleted. Important information is lost but grammatical correctness is maintained.

5 Experimental Results

The results data consisted of a set of 83 documents reporting soccer matches. The information in these texts is presented in short segments containing two or three sentences. In total, the data set consisted of 1087 sentences that included 2287 clause boundary candidates. The sentences were analyzed to determine if the ambiguity of the candidate use could be resolved. Initially the system recognized 1822 (79.7%) of the candidates that are clause boundaries. The remaining 465 (20.3%) candidates were used in another manner. The application of the clause resolution rules improved these figures to 1786 (92.3%) correct boundary identifications from 1934 instances. This improvement was achieved with only 2.0% of the correct boundaries omitted from the analysis. This loss is not only minimal but does not significantly affect the system's

performance. The 148 non-boundary candidates affect few sentences as only sentences with clauses of greater similarity than the complete sentence provoke the splitting process. In these examples, the majority of sentences were correctly processed.

Table 2 Categorization of sentences found in the sample data of 1087 sentences.

Sentence Type	Correct	Incorrect	Total
Lists	10	0	10
Quotes	10	0	10
Simple (no candidate)	110	0	110
Simple (candidates)	72	9	81
Compound / Complex	318	160	478
Other Sentences	249	149	398
TOTAL	749	318	1087

In the set of 1087 sentences, 17.6% are simple sentences. Of these, 57.6% (110) did not contain a candidate and 79.1% of the remainder were identified correctly. There are 478 sentences containing two clauses, excluding centric non-restrictive clauses, of which 66.5% were correctly identified. Over one third (308) of the sentences either contain more than two clauses or a central non-restrictive clause, of which 91.1% were correctly identified. In total 68.9% of the sentences analyzed were categorized correctly.

We have also conducted a small series of experiments on a subset of documents from the DUC 2004 corpus. The preliminary results suggest that the non-domain specific rules are valid for these texts, although there are systematic differences in the use of other syntactic categories, other than nouns, surrounding the candidate term “and” that extend rule 4.

6 Conclusion

Detecting correct clause boundaries was the primary goal of the work reported here. The system’s effectiveness in achieving this was reasonably high considering the complex and unconventional nature of web text. Superfluous use of commas and the incorrect use of the conjunction “and” greatly contribute to these problems. The removal of additional text within subordinate clauses is only complicated by the omission of the initial subordinating conjunction introducing the clause when the subordinate clause precedes the main clause. Non-restrictive clauses bounded by commas also cause poorer clause recognition. Although the use of conjunctions following both punctuation characters signifies normal clause boundaries, there appears no reliable method to recognize comma-bound non-restrictive clauses placed within a sentence.

A sentence cannot be assumed to be a single unit of information. Splitting text of any structure involves consideration of text coherence and context, and is therefore problematic. Nevertheless a method of accurately identifying tokens of clause level units is advantageous to several NLP tasks such as summarization and duplicate text removal. The hurdle of merging the clauses from all the types of sentences is the focus of future work. The integration of theme text from several segments

complicates the question of which texts to remove and the question of relationships to other texts. This complicates the process of merging multiple clauses. The solution of this problem will allow for the complete integration of the extracted text in the Mediation Information System. The integrated version is later transformed to other media, such as speech, through XML style sheets. This allows for the production of extracted web text to a variety of media achieving the goal of system extensibility.

References

1. Yan T., and Garcia-Molina H., 1995. Duplication Removal in Information Dissemination, In *Proc of VLDB-95, pp66-77*, September. 1995
2. Satoshi S., Chikashi N., 1997. Sentence Extraction and Information Extraction technique, *Document Understanding Conference 2003.*
3. Seki Y., 2003 Sentence Extraction by tf/idf and Position Weighting from Newspaper Articles, *Document Understanding Conference 2003.*
4. Klaus, Z 1997. A Literature Survey on Information Extraction and Text Summarization, Carnegie Mellon University, April 1997
5. DUC 2003, Document Understanding Conference 2003, http://www-nlpir.nist.gov/projects/duc/
6. Barzilay R., Elhadad N., and McKeown K., 2002. Inferring Strategies for Sentence Ordering in Multidocument News Summarization *JAIR 17, pp35-55.*
7. Nenkova, A., Schiffman B., Schlaiker A., Blair-Goldensohn S., Barzilay R., Sigelman S., Hatzivassiloglou V., McKeown K. 2003, Columbia University at the Document Understanding Conference 2003.
8. Goldensohn S., Evans D., Hatzivassiloglou V., McKeown K., Nenkova A., Passonneau, R., Schiffman B., Schlaikjar A., Siddharthan A., Siegelman S., 2004. Columbia University at DUC 2004, *Document Understanding Conference.*
9. Barzilay R., McKeown K., and Elhadad N., 1999. Information Fusion in the Context of Multi-Document Summarization. *ACL* 1999, pp703-733.
10. Lyons, S. and Smith, D., 2002. Domain-Specific Information Extraction Structures, *DEXA Workshops 2002: 80-84*
11. Tjong E.F. and Déjean H., 2001, Introduction to the CONLL-2001 Shared Task: Clause Identification, *CoNLL-2001*. http://cnts.uia.ac.be/conll2001/clauses/
12. Carreras, X. and Màrquez, L., 2001. Boosting Trees for Clause Splitting. In *CoNLL'01, 5th International Conference on Computational Natural Language Learning*, Toulouse, France
13. Carreras X., Màrquez L., Evans V., and Roth D., 2002 Learning and Inference for Clause Identification. *ECML*, Finland 2002
14. Mitkov R., Evans R., Orasan C., Barbu C., Jones L., Sotirova V., 2000. Coreference and anaphor: developing annotating resources and annotation strategies, *DAARC2000*, 49-58
15. Ginker M., 1994. *Clauses: Restrictive and Nonrestrictiv*e http://www.kentlaw.edu/academics/lrw/grinker/LwtaClauses__Restrictive_and_Nonrest.htm