Term-based identification of sentences for text summarisation

Byron Georgantopoulos^{*®} and Stelios Piperidis^{*&}

* Institute for Language and Speech Processing Epidavrou & Artemidos 6, 151 25 Maroussi, Greece

(@) University of Athens

(&) National Technical University of Athens

Email: {byron, spip}@ilsp.gr

Abstract

The present paper describes a methodology for automatic text summarisation of Greek texts which combines terminology extraction and sentence spotting. Since generating abstracts has proven a hard NLP task of questionable effectiveness, the paper focuses on the production of a special kind of abstracts, called *extracts:* sets of sentences taken from the original text. These sentences are selected on the basis of the amount of information they carry about the subject content. The proposed, corpus-based and statistical approach exploits several heuristics to determine the summary-worthiness of sentences. It actually uses statistical occurrences of terms (TF- IDF formula) and several cue phrases to calculate sentence weights and then extract the top scoring sentences which form the extract.

1. Introduction

Text summarisation is of great interest nowadays, where huge volumes of texts are produced and published electronically, resulting in new requirements for their management and processing (terminology extraction, text classification, information retrieval, information extraction, automatic abstracting etc.). The number of papers that are published today is ever increasing, while the Internet burst has created vast libraries of machinereadable texts. For one to keep informed and updated about the recent advances in his field has become now both vital and difficult. Since it is impossible to read through all the papers which are published nowadays, it is of great help to present them in a condensed way, i.e. using abstracts which summarise the content of an article. In this way the reader can get quickly a general idea about the article and decide if it is interesting enough to read it all through.

However, the work of abstracting a document is far from being easy. It requires skilled and specialised abstractors and of course it takes plenty of time. In the same way as machine translation, linguists and computer scientists have tried in recent years to substitute and/or aid human with machine abstractors. Automatic (or computerbased) abstracting, has shown considerable progress, has created several techniques and theories and has even produced some commercial software.

In this paper we focus on the production of a special kind of abstracts, called extracts: sets of sentences taken from the original text. These sentences are selected on the basis of the amount of information they carry about the subject content. Using statistical tools, the system "learns" from a corpus of papers which elements of a sentence make it important enough to function as a highly representative sentence. There are several such elements explored in the existing bibliography, the system utilises four of them: terms, cue phrases and sentence length. The structure of the paper is as follows: the first part of the paper is devoted to automatically locating terms, which then are fed to the sentence selection mechanism. The second part deals with the scoring of sentences and the creation of the extract.

2. Term Extraction

The method aims at linguistically processing machinereadable text corpora and extracting lists of candidate single and multi-word terms of the domain. A term is a linguistic realisation of a domain specific concept and usually is lexicalised in the form of a noun phrase. In bibliography, one can find two basic methods for extracting terms:

- 1. Using a term grammar (usually a context free grammar) which is applied to an appropriately annotated text and extracts all the phrases it recognises (Bourigault, 1992)
- 2. Using statistical tools similar to the ones developed in the field of information retrieval and text indexing. These tools include frequency counting, formulas from information theory, formulas which take into account the context of words, etc. (Daille B. 1994; Frantzi and Ananiadou 1997)

There are important differences between these two lines of action. A term grammar describes the syntactic structure that a valid term must satisfy, but it is possible that phrases recognised by the grammar are not valid terms. The weakness of a grammar is attributed to the fact that its rules, though a subset of NP rules, are general enough to generate a large number of potential terms¹. Furthermore, a grammar cannot locate single word terms since such a term does not have any syntactic structure

¹ For example, the typical rule Term ::- Adj+Noun will definitely recognise many non-term phrases besides valid ones.

except part-of-speech information². In general, a term grammar can only produce a set of potential terms that remain to be validated by an expert or a module of different nature.

The statistical approach is based on the assumption that words and phrases indicative of the domain of a document tend to appear *frequently* (the same applies for phrases consisting of words that appear frequently together). Frequency can have two different interpretations: (1) a phrase is more frequent in the current text than in a representative collection of texts belonging to its domain and (2) a phrase is more frequent than others in the same text. Based on this "competitive" conception of frequency, each phrase is assigned a score representing its significance, (not taking into account functional words). Phrases at the top of this ranking have the highest probability of being valid terms. This method can extract single-word terms as well as multi-word terms. On the other hand, it cannot locate terms which do not satisfy the statistical criteria, i.e. they are not frequent enough. This is partly due to the fact that it is difficult to draw the line between middle frequency and high frequency. Finally, the selected statistical formula can affect the performance of extraction in the same way that the selected rules of the grammar, i.e. its syntactical coverage, affect the performance of the grammatical method.

In between these methodologies stand other approaches which combine statistical processing with linguistic modelling (Daille B. 1994; Frantzi and Ananiadou 1997; Georgantopoulos and Piperidis 1998). These hybrid systems initially construct a candidate term list using a term grammar and then filter this set through statistical techniques in order to remove syntactically acceptable phrases that are not "frequent" enough to be assigned valid termhood.

The term extraction process in the proposed method is a hybrid one, and operates in three pipelined stages:

- 1. morphosyntactic annotation of the domain corpus (including below part-of-speech tagging information)
- 2. corpus parsing based on a term pattern grammar endowed with regular expressions and featurestructure unification
- 3. statistical filtering in order to remove grammarextracted terms lacking statistical evidence

The following diagram illustrates the processing stages of term extraction:





2.1. Grammar Parsing

The pattern grammar³ used in the syntactic analysis is a subset of pattern rules presented in (Gavriilidou and Lambropoulou 1994), whose rules cover a great part of the Greek terminology. It also utilises feature structure unification formalism (typical in grammar theories like HPSG) and regular expression operators. For example, the pattern that describes terms of the form: NOUN PREPOSITION (ART?) NOUN has the following format:

```
Term pattern : (cat = Noun
              (cat = Pronoun)
              type = Cl),
             [ (cat = Prep
                type = Sp);
                ^{(cat = Art)}
                  gender = G
                  number = N
                  case = C)];
              (cat = Prep
              type = Pa
              gender = G
              number = N
              case = C],
              (cat = Noun
              gender = G
              number = N
              case = C)
             ).
```

The '^' symbol at the end denotes optionality (zero or one appearance), the ';' symbol is the 'OR' operator and brackets are used to group elements. The basic constraint posed by this rule is the number-case-gender agreement between nouns and articles.

The term grammar consists of rules recognising two to four-word terms⁴. Each rule was converted to a nondeterministic finite state automaton (NDFA). NDFA's were used in preference to context-free grammar parsers (like Prolog DCG) because (a) they are much faster, operating in linear time (b) typical parsers do not support regular operators directly. Features used in unification include grammatical category as well as subcategorisation

³ We use the terms, *pattern grammar* and *term grammar* interchangeably

⁴ By words, in this context, we refer to content words.

features like gender, case, tense, number, etc. Typical regular expression operators are optionality, kleene star, disjunction, etc. Such a non-deterministic automaton is illustrated below:



Grammar rule :- Noun Art[G,N,ge]? Noun[G,N,ge] Conj[co] Art[G1,N1,ge]? Noun[G1,N1,ge]

2.2. Statistical Filtering

After the term grammar module has been applied, the extracted terms are statistically evaluated in order to remove items without adequate statistical evidence and thus. reduce the overgeneration effect caused by pattern grammars. Statistical evaluation is performed using TFIDF (Term-Frequency Inverse-Document-Frequency (Salton et al. 1989), so that the frequency of the term in the domain is also taken into consideration. Only the top-ranked terms are extracted, thus reducing a lot the noise introduced by the pattern grammar. TFIDF is a standard weight computation method which combines term frequency (TF_i), the number of documents (N) and the number of documents (n_i) that the term appears in:

weight
$$i = TF_i \cdot \log \frac{N}{n_i}$$

The TFIDF scoring formula favours terms which are highly frequent in a document but rare in the rest of the corpus. There are several TFIDF weighting schemata, since both TF and IDF can be parameterised:

Term Frequency		Inverse Document Frequency	
<i>n</i> (natural) <i>I</i> logarithmic)	$ \begin{array}{c} tf \\ 1 + \ln(tf) \end{array} $	<i>n</i> (no)	$l \log \frac{N}{N}$
a(augmented)	$0.5+0.5 \frac{tf}{\max tf}$	<i>J</i> (1011)	" ni

Each formula is thus abbreviated by two letters: for example, the previous formula's abbreviation is: *nf*.

3. Sentence Extraction

Techniques for automatic text summarising today fall under two general categories:

1. Automatic abstracting through text understanding: This method is very close to the way humans do abstracting: the system has first to understand the text and then to create the abstract. The first step produces a canonical-logical form of the text that feeds the sentence generator to write the abstract. Both of these issues, text analysis and understanding, and text generation are important NLP problems by themselves, and have proved computationally difficult to achieve so far. The complexities of language (anaphora, context, polysemy, global common knowledge required, etc.) make it too hard to create a system which will be effective in terms of producing good summaries, as well as being fast and maintainable.

2. Automatic summarising via sentence extraction: This method aims at locating the best content-bearing sentences in a text. Extracting sentences is a much simpler and hence a fast and feasible approach. The assumption behind extracting is that there should be a set of sentences which present all the key ideas of the text, or at least a great number of these ideas. The goal is first to identify what really influences the significance of a sentence, what makes it important. The next step is to train and program a system to automatically locate these elements in a sentence and compute its summary-value. It is evident that this method avoids all the above-mentioned conventional NLP problems: no analysis, representation and understanding of the text is required. Also, no generation has to take place, and in addition the extracted sentences will be perfectly grammatical. On the other hand the resulting passage might not be much comprehensible and refined: coherence is sacrificed for speed and feasibility.

Summary-worthy sentences should be selected on the basis of how well they represent the subject content. This merit of representativeness has an arithmetic value (score) and it depends on its diagnostic units. A diagnostic unit is anything in a sentence which gives a clue to its significance. It can be a word, a sequence of words, a syntactic structure, the sentence length, its position within the text, special formatting, etc. The scoring formula can encapsulate more than one diagnostic units and has the form of a weighted sum:

$$score = \frac{1}{L} \sum_{i=1}^{n} a_{w_i} h_i$$

where a_{wi} is the weight of the i-th heuristic, h_i is the specific score of the sentence for the i-th heuristic, n is of course the number of heuristics used and L is the sentence length. The implemented system takes into account three major heuristics:

- 1. Terms: term weights are determined as in the termextraction mode
- 2. Cut-off length: sentences below the specified length receive a zero-score, based on the assumption that very small sentences are not usually summary-worthy
- 3. Cue phrases: It is typical in paper writing to use certain words or phrases to highlight sentences.

Phrases such as *This paper ..., we propose ..., important* are frequently indicative of a worthy sentence for summarisation. If a sentence contains any of a cue phrase (user defined, along with their scores) then its score is altered accordingly. Each cue phrase is associated with a score (positive or negative).

The following diagram illustrates the algorithm for computing each sentence's score:



After each sentence has been scored, the top-ranked sentences (by absolute number or by percentage) are extracted and presented in the original order they appear in the text.

Our immediate future plans include:

- Improving the efficiency of the term extraction module by reducing the number of potential terms recognised by the grammar module and improving its coverage. To this end we intend to:

(a) Utilise further syntactic information (NP head) in order to group together terms with the same semantic but slightly different syntactic structure (b) Extend an already existing terminological base through linguistic operations such as overcomposition, modification, coordination, etc.

- Regarding the sentence extraction module, and in order to improve the coherence [balance] of the extract, we investigate methods to compute similarity between the extracted sentences in order to remove semantically similar sentences for evaluation purposes. We plan to create a golden corpus of target extracts in order to measure how well the computer "simulates" humans in pinpointing the important sentences in a text.

References

- Association for Computational Linguistics 1997. Proceedings of a Workshop on Intelligent, Scalable Text Summarization, Madrid, Spain.
- Bourigault D. 1992. Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases. *Proceedings of the 14th International Conference on Computational Linguistics.*

- Daille B. 1994. Study and implementation of combined techniques for automatic extraction of Terminology. *In The Balancing Act: Combining Symbolic and Statistical Approaches to Languages*, Workshop at the 32nd Annual Meeting of ACL, Las Cruces, Nouveau Mexique.
- Edmundson, H. P. 1969. New methods in automatic extracting, *Journal of the Association for Computing Machinery* 16(2), 264--285.
- Frantzi, K.T. and Ananiadou, S. 1997. Automatic term recognition using contextual clues, *In Proceedings of Mulsaic* 97, IJCAI, Japan.
- Gavriilidou M, Lambropoulou P. 1994. *Report on the Constituent Grammar*, RENOS project, LREI- 62-048, Athens
- Georgantopoulos B., Piperidis S. 1998. Automatic acquisition of terminological resources for information extraction applications. *In Proceedings of the 1st Panhellenic Conference on New Information Technologies*, pp 279–287, Athens.
- Kupiec, J., Pedersen, J. Chen, F. 1995. A trainable document summarizer, *In Proceedings of the Eighteenth Annual International ACM-SIGIR '95*, pp. 68--73.
- Paice, C. D. 1990. Constructing literature abstracts by computer: Techniques and prospects, *Information Processing Management* 26(1), 171--186.
- Proceedings of the AAAI'98 Spring Symposium on Intelligent Text Summarization, Stanford, March 1998. Copyright AAAI.
- Salton, G. 1989. Automatic text processing: the transformation, analysis, and retrieval of information by computer, Reading, Mass. Wokingham : Addison-Wesley.
- SUMMAC 1999. Results of TIPSTER Text Summarization Evaluation Conference (SUMMAC).
- Teufel S. and Moens M 1998. Sentence extraction and rhetorical classification for flexible abstracts. *In AAAI Spring Symposium on Intelligent Text Summarization*.