Creating Summarization Systems with SUMMA

Horacio Saggion
Department of Information and Communication Technologies
Universitat Pompeu Fabra
C/Tanger 122
Barcelona - 08018
Spain
horacio.saggion@upf.edu

Abstract

Automatic text summarization, the reduction of a text to its essential content is fundamental for an on-line information society. Although many summarization algorithms exist, there are few tools or infrastructures providing capabilities for developing summarization applications. This paper presents a new version of SUMMA, a text summarization toolkit for the development of adaptive summarization applications. SUMMA includes algorithms for computation of various sentence relevance features and functionality for single and multidocument summarization in various languages. It also offers methods for content-based evaluation of summaries.

Keywords: Text Summarization Tools, Applications, Infrastructure

1. Introduction

Due to the overabundance of textual information on-line, automatic text summarization, the reduction of a text to its essential content (Saggion and Poibeau, 2013) is fundamental for information systems dealing with textual content. In recent years text summarization research has intensified with well known evaluation programmes promoting the interest in the area (Owczarzak and Dang, 2010; Over et al., 2007; Giannakopoulos et al., 2011; Giannakopoulos and Petasis, 2013). In this context, comparison of different summarization approaches, which is usually done by relying on well established or widely used or accessible systems such as MEAD (Radev et al., 2004), is of paramount importance in text summarization. However, MEAD only provides few summarization functionalities or features such as a position-based feature, a centroid-based feature, and a first-sentence similarity feature which could be limited for comparison purposes. Availability of customizable Natural Language Processing (NLP) systems, and not only ready-made applications, is also important for teaching NLP at university. In this paper we present a new and improved version of the SUMMA toolkit (Saggion, 2008), a library and GATE plug-in (Maynard et al., 2002) which can be used to implement different summarization solutions, making it ideal for the creation of baseline and advanced summarization systems. SUMMA can also be used as a teaching aid in NLP, in fact it is being used as a teaching tool at Universitat Pompeu Fabra to demonstrate theoretical and practical concepts in a NLP course and to facilitate the development of NLP laboratory assignments.

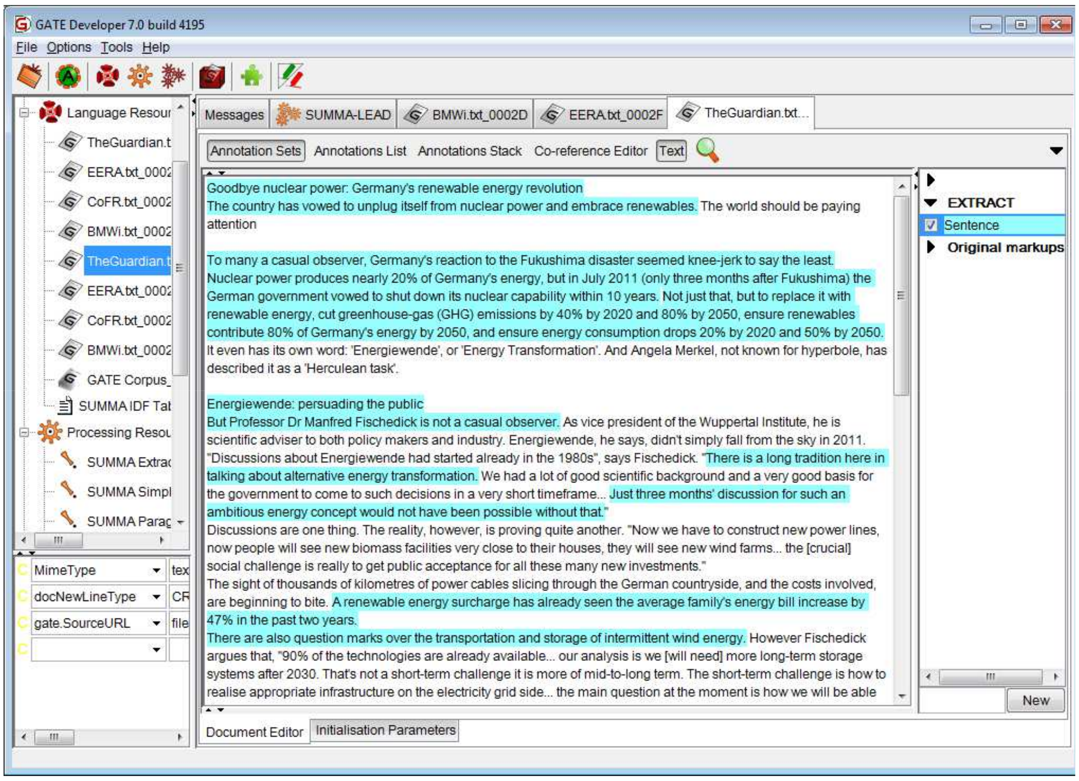
2. Related Work

Research on text summarization has progressed steadily in recent years, however, and contrary to what occurs with NLP tools such as parsers, named entity recognizers, POS taggers, etc. which can be obtained from well known packages such as GATE[1], OpenNLP[2], Stanford CoreNLP[3], etc. there are no many softwares providing summarization capabilities. There are many text summarization systems reported in the literature (Mihalcea and Tarau, 2004; Erkan and Radev, 2004; Plaza et al., 2008; Litvak and Last, 2008), however there are few systems which allow one to *create* text summarization solutions adapted to new domains, text types, or genres. The MEAD system (Radev et al., 2004) is a well known example of a tool which has been widely used for comparison purposes, however it has fewer components and customization possibilities than SUMMA. Compendium (Lloret, 2012) is a text summarization architecture which includes components for relevance computation and redundancy removal, however apart from an on-line demo, it is not freely available. Several systems used in international evaluation programs – Cortex (Boudin and Moreno, 2007) or Classy (Conroy et al., 2005), etc. – are generally used locally within an specific organization.

3. Refining SUMMA

SUMMA is a set of text summarization resources for the development of text summarization applications. It has been implemented using the GATE API and relies on GATE documents, GATE annotations, and other data-structures for information storage and retrieval. Since 2010, several improvements have been made to SUMMA including the development of new components and better parametrization of existing ones. SUMMA has been used since 2010 in different projects and evaluation programmes including the TOPAS Project[4] (patent summarization in English, French and German), the INEX evaluations [5], the DEFT evalua-

[1] `http://gate.ac.uk`
[2] `https://opennlp.apache.org`
[3] `http://nlp.stanford.edu/software/corenlp.shtml`
[4] `http://topasproject.eu/index.php`
[5] `https://inex.mmci.uni-saarland.de/`



Goodbye nuclear power: Germany's renewable energy revolution
The country has vowed to unplug itself from nuclear power and embrace renewables. The world should be paying attention

To many a casual observer, Germany's reaction to the Fukushima disaster seemed knee-jerk to say the least. Nuclear power produces nearly 20% of Germany's energy, but in July 2011 (only three months after Fukushima) the German government vowed to shut down its nuclear capability within 10 years. Not just that, but to replace it with renewable energy, cut greenhouse-gas (GHG) emissions by 40% by 2020 and 80% by 2050, ensure renewables contribute 80% of Germany's energy by 2050, and ensure energy consumption drops 20% by 2020 and 50% by 2050. It even has its own word: 'Energiewende', or 'Energy Transformation'. And Angela Merkel, not known for hyperbole, has described it as a 'Herculean task'.

Energiewende: persuading the public
But Professor Dr Manfred Fischedick is not a casual observer. As vice president of the Wuppertal Institute, he is scientific adviser to both policy makers and industry. Energiewende, he says, didn't simply fall from the sky in 2011. "Discussions about Energiewende had started already in the 1980s", says Fischedick. "There is a long tradition here in talking about alternative energy transformation. We had a lot of good scientific background and a very good basis for the government to come to such decisions in a very short timeframe... Just three months' discussion for such an ambitious energy concept would not have been possible without that."
Discussions are one thing. The reality, however, is proving quite another. "Now we have to construct new power lines, now people will see new biomass facilities very close to their houses, they will see new wind farms... the [crucial] social challenge is really to get public acceptance for all these many new investments."
The sight of thousands of kilometres of power cables slicing through the German countryside, and the costs involved, are beginning to bite. A renewable energy surcharge has already seen the average family's energy bill increase by 47% in the past two years.
There are also question marks over the transportation and storage of intermittent wind energy. However Fischedick argues that, "90% of the technologies are already available... our analysis is we [will need] more long-term storage systems after 2030. That's not a short-term challenge it is more of mid-to-long term. The short-term challenge is how to realise appropriate infrastructure on the electricity grid side... the main question at the moment is how we will be able

Figure 1: Highlighted Sentences in Document.

tions[6], and TAC [7]. It is currently being used in the Multi-Sensor[8] project and adapted for the Dr. Inventor project[9]. It has also been used for comparative evaluation in different research contexts (e.g., (Plaza, 2011), (Lloret, 2012) and (Aker et al., 2013)).

3.1. Processing Resources

SUMMA contains over twenty processing resources (i.e., implemented algorithms in GATE jargon) which are fully described in the SUMMA website [10]. Most processing resources require tokens (e.g. words) and sentences which can be easily obtained applying the GATE default tokeniser and sentence splitter. Apart from these basic units, some components will also require the computation of named entities, which can also be produced by applying the off-the-shelf ANNIE system to the input documents. Processing resources add feature-values to existing annotations or create new annotation types such as vector representations or n-grams. Summaries in SUMMA are annotations added to the document (see Figure 1). This makes SUMMA flexible enough to allow extraction of units of different granularities such as sentences, paragraphs, or sub-sentential units.

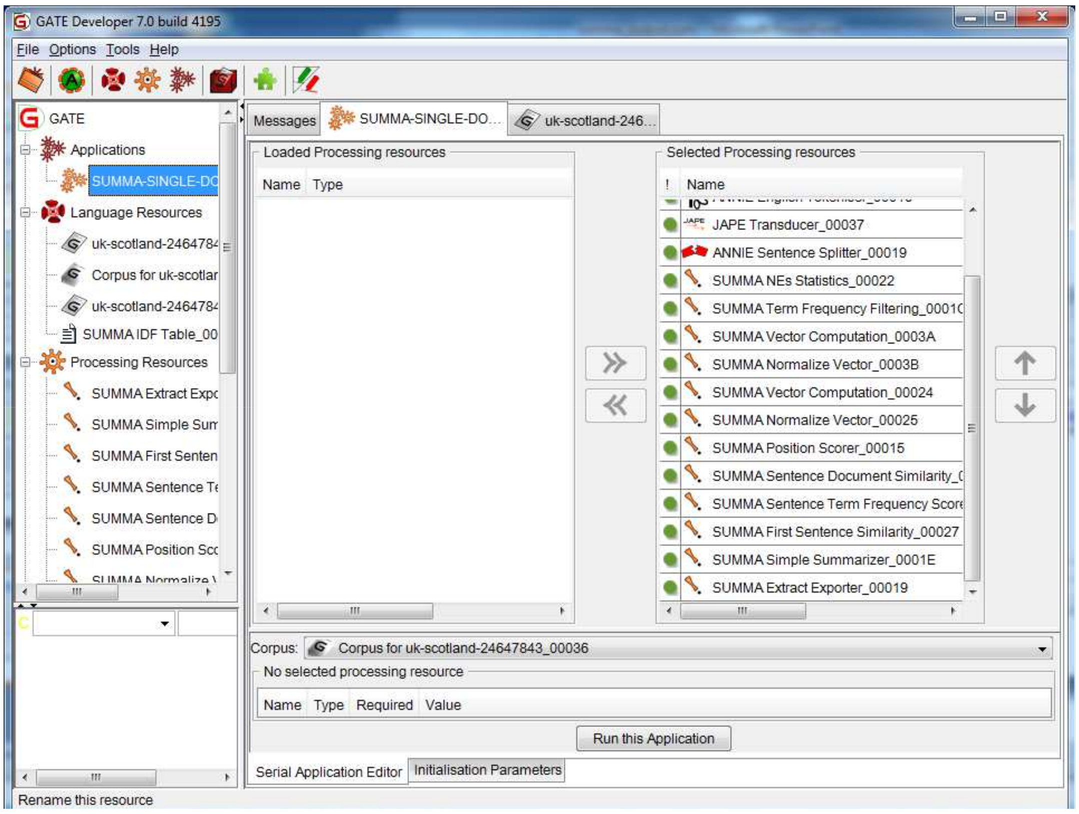
[6]http://deft2011.limsi.fr/
[7]http://users.iit.demokritos.gr/~ggianna/TAC2011/MultiLing2011.html
[8]http://www.multisensorproject.eu/
[9]http://drinventor.eu/
[10]http://www.taln.upf.edu/pages/summa.upf/

Some of the most relevant components for creating summarization applications are the following:

- *SUMMA NEs Statistics*. It computes statistics – term frequency and *tf*idf* statistics – for words in the input document. These statistics are stored as features of the words. The component needs as a parameter an inverted document frequency table also available in the toolkit (see Section 3.2.).

- *SUMMA Vector Computation*. This module creates vector representations – implemented as feature maps – which contain terms (e.g. words) with associated numerical values representing the weight of the terms. The module is designed to create representations for different document parts such as sentences, paragraphs, titles, full sections or the full document. Term weights can be customized to be frequencies, or *tf*idf* values, etc. Figure 3 shows a sentence vector computed in a document. In order to compact the representations, stop word tables can be specified to filter out stop words or specific types of tokens.

- *SUMMA Title Sentence Similarity*. Although its name could indicate the implementation of the title method (Edmundson, 1969) in text summarization, it in fact allows us to compare each sentence in the document to any other textual unit in the same document (e.g. the title) producing a similarity value which we can use as a relevance measure. In order to produce such



GATE
Applications
SUMMA-SINGLE-DC
Language Resources
uk-scotland-2464784
Corpus for uk-scotlar
uk-scotland-2464784
SUMMA IDF Table_00
Processing Resources
SUMMA Extract Expc
SUMMA Simple Sum
SUMMA First Senten
SUMMA Sentence Te
SUMMA Sentence D
SUMMA Position Scc
SUMMA Normalize V

Loaded Processing resources

Name	Type

Selected Processing resources

!	Name
●	ANNIE English Tokeniser_00015
●	JAPE Transducer_00037
●	ANNIE Sentence Splitter_00019
●	SUMMA NEs Statistics_00022
●	SUMMA Term Frequency Filtering_0001C
●	SUMMA Vector Computation_0003A
●	SUMMA Normalize Vector_0003B
●	SUMMA Vector Computation_00024
●	SUMMA Normalize Vector_00025
●	SUMMA Position Scorer_00015
●	SUMMA Sentence Document Similarity_(
●	SUMMA Sentence Term Frequency Score
●	SUMMA First Sentence Similarity_00027
●	SUMMA Simple Summarizer_0001E
●	SUMMA Extract Exporter_00019

Corpus: Corpus for uk-scotland-24647843_00036

No selected processing resource

Name	Type	Required	Value

Figure 2: Summarization Application in the GATE GUI.

value we apply the *cosine* formula to the vector representations of the sentence and the target textual unit. This component requires vector representations for the sentences and the target unit to compare the sentences to.

- *SUMMA Position Scorer*. This component is an implementation of the classical position feature (Lin and Hovy, 1997) in which sentences receive a relevance based on their positions in the document. Moreover, the *SUMMA Paragraph Scorer* module can be used to assess sentence relevance according to the location of the sentence in its paragraph. The weights the sentences receive can be customized.

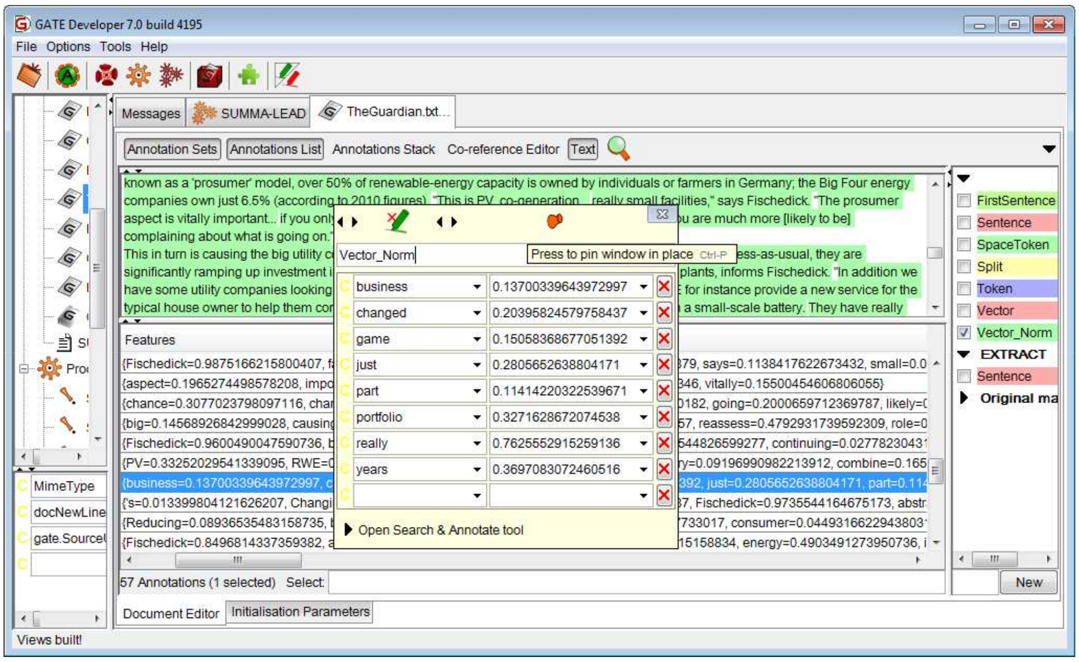
- *SUMMA Cue Phrase Scorer*. This resource is a cue-based relevance feature computation procedure (Paice, 1990). In order to use it, a cue-phrase list which contains ”important” words and their weights is needed (see 3.2.). The weight of the sentence will be proportional to the sum of weights of the important words it contains.

- *SUMMA Query Method Scorer*. This component implements a query-based summarization feature which scores sentences based on the similarity of a sentence to an input query (Tombros et al., 1998; Saggion et al., 2003). The query – a parameter of the component – is in fact a document which should have a single vector annotation representing the whole query; this query vector is compared to each sentence vector in the input document using the cosine function to produce a relevance feature.

- *SUMMA Term Frequency Scorer*. This component implements a relevance scoring function based on weights associated to the terms in the sentences, these could be *tf*idf* or any other statistic. A normalized relevance is produced after all sentences have been processed in order to obtain relevance values in the interval $(0..1)$.

- *SUMMA Semantic Scorer* implements a relevance feature for the sentence that is proportional to the number of named entities and types the sentence contains. The types of the named entities to be considered can be customized.

- *SUMMA N-Gram Computation*. This component implements customized n-grams which are necessary to support content-based evaluation in ROUGE (Lin, 2004). Several parameters can be used to decide the annotation (e.g. words, non-stop words, nouns) and annotation feature (e.g. word itself, lemma, stem) used to compute the n-grams as well as the size of the n-gram. This parametrization makes it possible to replicate various conditions available in the ROUGE package.



```
known as a 'prosumer' model, over 50% of renewable-energy capacity is owned by individuals or farmers in Germany; the Big Four energy
companies own just 6.5% (according to 2010 figures). "This is PV, co-generation ... really small facilities," says Fischedick. "The prosumer
aspect is vitally important... if you only                                    ou are much more [likely to be]
complaining about what is going on."
This in turn is causing the big utility c                                     ess-as-usual, they are
significantly ramping up investment i                                         plants, informs Fischedick. "In addition we
have some utility companies looking                                           for instance provide a new service for the
typical house owner to help them co                                           a small-scale battery. They have really
```

Vector_Norm

business	0.13700339643972997
changed	0.20395824579758437
game	0.15058368677051392
just	0.2805652638804171
part	0.11414220322539671
portfolio	0.3271628672074538
really	0.7625552915259136
years	0.3697083072460516

```
Features
{Fischedick=0.9875166215800407, f            379, says=0.1138417622673432, small=0.0
{aspect=0.1965274498578208, impo             346, vitally=0.15500454606806055}
{chance=0.3077023798097116, char             0182, going=0.2000659712369787, likely=0
{big=0.14568926842999028, causing            57, reassess=0.4792931739592309, role=0
{Fischedick=0.9600490047590736, b            544826599277, continuing=0.0277823043
{PV=0.3325202954133905, RWE=0                ry=0.09196990982213912, combine=0.165
{business=0.13700339643972997, c             392, just=0.2805652638804171, part=0.114
{'s=0.013399804121626207, Changi             37, Fischedick=0.9735544164675173, abstr
{Reducing=0.08936535483158735, b             733017, consumer=0.0449316622943803
{Fischedick=0.8496814337359382, a            15158834, energy=0.4903491273950736, i
57 Annotations (1 selected)   Select:
```

Figure 3: Vectors in a Document Processed by SUMMA.

- *SUMMA Centroid*. In order to support multi-document summarization functionalities SUMMA implements a *centroid* computation procedure for a set of documents (i.e., a corpus in GATE) (Saggion and Gaizauskas, 2004). The centroid, which is the average of all document vectors in the corpus is stored for further processing.

- *SUMMA Sentence Centroid Similarity*. This module computes the similarity of each sentence in a document to the centroid of a set of documents. Here again, the comparison is based on the cosine function.

- *SUMMA Simple Summarizer* is a module which performs the final sentence scoring computation. It can be customized to indicate which features and associated weights will be used to score sentences. This component is also in charge of selecting the top ranked sentences according to the compression parameter, create the annotations for the summary, and eventually create an independent document for the generated summary. A flag can be used to score sentences only which is relevant for multi-document summarization.

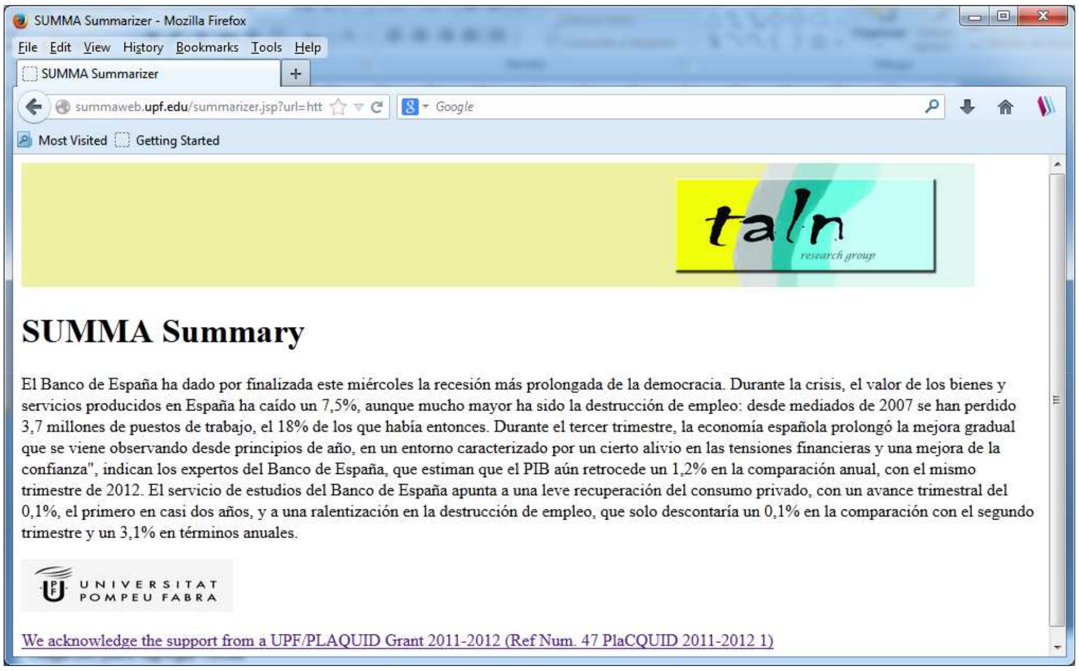
- *SUMMA Simple Multi-document Summarizer* has similar functionalities to the above, but it operates over a set of related documents which sentences have already being scored. It selects top ranked sentences from the set of documents filtering out redundancy. The sentences selected from each document are annotated and a new document created with them. To avoid incoherence all sentences from the same document will appear together in the multi-document summary. The order in which the sentences appear in the summary will be that of the sentences in the input documents.

Some additional resources include a summary exporter to save summarization results to disk, a random summarizer, a vanilla multi-document summarizer, a term filtering module, and a random summarizer. Figure 2 shows SUMMA components pipelined in a summarization application in the GATE development environment.

3.2. Language Resources

SUMMA implements language resources needed by several of its components. For example in order to perform statistical analysis, SUMMA relies on inverse document frequency (*idf*) tables (Salton and McGill, 1983) which can be created from external resources or directly with a SUMMA component. The table of inverted document frequencies is a text file which first line contains the number of documents (num_docs) used to compute the idf statistics, and then each line contains a word (w) from the corpus together with the number of documents where the word occurs (count(w)). Below is an example of idf table for Spanish:

```
298
atentados 4
```


SUMMA Summarizer - Mozilla Firefox
File Edit View History Bookmarks Tools Help
SUMMA Summarizer
summaweb.upf.edu/summarizer.jsp?url=htt
Google
Most Visited Getting Started
taln research group

SUMMA Summary

El Banco de España ha dado por finalizada este miércoles la recesión más prolongada de la democracia. Durante la crisis, el valor de los bienes y servicios producidos en España ha caído un 7,5%, aunque mucho mayor ha sido la destrucción de empleo: desde mediados de 2007 se han perdido 3,7 millones de puestos de trabajo, el 18% de los que había entonces. Durante el tercer trimestre, la economía española prolongó la mejora gradual que se viene observando desde principios de año, en un entorno caracterizado por un cierto alivio en las tensiones financieras y una mejora de la confianza", indican los expertos del Banco de España, que estiman que el PIB aún retrocede un 1,2% en la comparación anual, con el mismo trimestre de 2012. El servicio de estudios del Banco de España apunta a una leve recuperación del consumo privado, con un avance trimestral del 0,1%, el primero en casi dos años, y a una ralentización en la destrucción de empleo, que solo descontaría un 0,1% en la comparación con el segundo trimestre y un 3,1% en términos anuales.

UNIVERSITAT POMPEU FABRA

We acknowledge the support from a UPF/PLAQUID Grant 2011-2012 (Ref Num. 47 PlaCQUID 2011-2012 1)

Figure 4: Summarization Application on the Web (`http://summaweb.upf.edu/`).

```
abandonada 1
Margaret 2
sardina 1
delictivo 1
....
```

For idf computation of a word w we use the following standard formula:

$$idf(w) = \log_2(\frac{num\_docs}{count(w)})$$

The default value for words not appearing in the table is one. The table itself can be generated by any program, however SUMMA contains a module which can be used to produce such a table from a pre-processed corpus. This module allows the user to adapt the word characteristics – for example restricting the counts to: (i) the original words of the document; (ii) their lemmas; or (iii) any normalized version of the word (e.g. lowercased).

Tables for stop words and stop categories are also implemented to support term filtering functionalities which are needed to create more compact vector representations. They are files containing one word per line as shown below.

```
ser
estar
les
el
la
los
...
```

A table of stop categories may be used to filter out some specific words from vector representations such as numbers, symbols, or any other non-word.

In order to implement our cue phrase component, a ”cue” word lists has to be implemented relying on a Gazetteer lists implementation available in GATE. One example list in the precise format required by SUMMA is shown below:

```
presents:count=10
discusses:count=5
shows:count=5
....
```

The list contains the cue word or expression and a *count* feature together with a value used to weight words in the cue-based method.

4. Applications

There are various ways to use SUMMA in the development of practical applications. The typical way to proceed will be to use the GATE GUI as a development environment loading SUMMA as a plug-in to create an application with SUMMA components. The application could be invoked from a stand-alone Java program. Another way to use SUMMA is by importing the library in a Java programming environment in order to program with SUMMA by creating resources and applying them to input documents.

4.1. Web Applications

Summarization applications to support single document summarization of news articles in Spanish, English, and Catalan have been developed and deployed in a server. The interface can be seen at `http://summaweb.upf.edu/` and the result of summarizing a document in Spanish can be appreciated in Figure 4. This application can create standalone summaries or sentence highlights.

4.2. Customizable Command Line Apps

In addition to the basic building blocks to create NLP pipelines, two readymade applications implemented as script programs are made available for single and multi-document summarization. These applications can be easily customized by setting up a fixed number of parameters in the script files. Example input texts are also provided to test the applications.

4.3. Java Demonstrators

The distribution of SUMMA includes two stand-alone programs to start *programming* with SUMMA. One program is designed to help developers in the integration of processing resources from SUMMA in a corpus pipeline. A second program illustrates how a full GATE application which contains resources from SUMMA can be loaded and executed over a document in order to extract a summary from it. One point one has to keep in mind when developing an application with SUMMA is that documents may need to be pre-processed before they can be summarized. For example Web pages will usually need to be cleaned-up of non-textual material in order to obtain reasonable output. This means that additional logic is usually needed to extract and isolate the text from the document before a SUMMA application can summarize it.

5. Evaluation in SUMMA

The GATE system provides resources for the computation of inter-annotator agreement, precision, recall, and f-score which could be used to compare sentences selected by a summarization system to sentences selected by humans, however, automatic evaluation in summarization is usually reported in the literature in terms of values of content-based measures (Lin, 2004; Saggion et al., 2010). The ROUGE package is the *de facto* evaluation tool used when reporting evaluation in summarization research. SUMMA does not integrate the ROUGE package but it implements some of its functionalities. The module to compute ROUGE statistics in SUMMA is called *SUMMA Rouge Evaluation* and can be used to compare the sentences selected by a SUMMA summarizer (e.g. annotations) with a corpus of ideal abstracts provided as input. It can also be used to compare one input text (a system summary) to a set of ideal summaries. The component can be customized to select the size (e.g., 1, 2, 3, 4) and type (e.g., only words, only lemmas) of the n-gram to be used in the computation of ROUGE values.

6. Teaching Summarization

We have used SUMMA as a teaching tool to cover in practical sessions automatic text summarization. In the theoretical part of the course a number of classic sentence relevance features are introduced in addition to advanced summarization techniques, additionally several summarization applications on the Web are demonstrated including one developed with SUMMA. In the practical sessions we match theory with practice. The tool is first used in a tutorial where the students interact with the tool creating a summarization application using the GATE GUI. In this tutorial session they follow the teacher who indicates how the different component should be customized and pipelined in an application. After adding a new component to the application they test it and observe the new features added and/or annotations created. At the end of the tutorial they are able to create a simple but complete application. In a second session they are asked to create from scratch two "portable" summarization application, one for English and another one for Spanish. They are given precise instructions of how they should be customized (e.g., features to combine, compression). The applications which are appropriately saved are used in a standalone Java program to summarize English or Spanish input.

7. Software Distribution and Installation

The SUMMA software is distributed through a dedicated website at `http://www.taln.upf.edu/pages/summa.upf/`. The site includes all documentation on processing resources required to use the software as well as Java examples of how to create standalone applications. The installation procedure is very simple, it only requires the user to copy the distribution directory to disk. Within GATE the software can be loaded as a plug-in.

8. Closing Remarks

In this paper we have presented a new version of SUMMA, a library of summarization components freely available for research purposes. This new version has been improved by making the components customizable in as much as possible, additionally a website dedicated to its distribution, documentation, and support has been created. Before this new release, SUMMA was only available by requesting a copy of the library.
We have used it for developing basic and advanced summarization systems in different languages and for different summarization tasks such as generic or query focused summarization, single or multi-document summarization, patent summarization, etc. Baseline summarization systems are easy to create by using one of the many relevance summarization feature computation modules (e.g., position, term frequency, title similarity). SUMMA is easy to install and use provided the user has some experience with the GATE system. The SUMMA website offers full documentation and practical examples to start developing applications.

Acknowledgments

The author acknowledges partial support from project Dr. Inventor (FP7-ICT-2013.8.1 611383), programa Ramón y Cajal 2009 (RYC-2009-04291), project number TIN2012-38584-C06-03 from Ministerio de Economía y Competitividad, Secretaría de Estado de Investigación, Desarrollo e

Innovación, Spain, and from UPF/PlaQUID Grant number 47 (2011-2012).

9. References

Aker, A., Plaza, L., Lloret, E., and Gaizauskas, R. (2013). Multi-document summarization techniques for generating image descriptions: A comparative analysis. In Poibeau, T., Saggion, H., Piskorski, J., and Yangarber, R., editors, *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing, pages 299–320. Springer Berlin Heidelberg.

Boudin, F. and Moreno, J. M. T. (2007). Neo-cortex: A performant user-oriented multi-document summarization system. In Gelbukh, A. F., editor, *CICLing*, volume 4394 of *Lecture Notes in Computer Science*, pages 551–562. Springer.

Conroy, J. M., Stewart, J. G., and Schlesinger, J. D. (2005). Classy query-based multi-document summarization. In *In Proceedings of the Document Understanding Conference 2005 (DUC 2005) at HLT/EMNLP*.

Edmundson, H. (1969). New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285, April.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December.

Giannakopoulos, G. and Petasis, G. (2013). *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*. Association for Computational Linguistics, Sofia, Bulgaria, August.

Giannakopoulos, G., El-Haj, M., Favre, B., Litvak, M., Steinberger, J., and Varma, V. (2011). Tac2011 multiling pilot overview. In *TAC 2011 Workshop*, Gaithersburg, MD, USA. NIST.

Lin, C. and Hovy, E. (1997). Identifying Topics by Position. In *Fifth Conference on Applied Natural Language Processing*, pages 283–290. Association for Computational Linguistics, 31 March-3 April.

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Litvak, M. and Last, M. (2008). Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, MMIES '08, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lloret, E. (2012). *Text Summarisation based on Human Language Technologies and its Applications*. Ph.D. thesis, Universidad de Alicante, Spain.

Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K., and Wilks, Y. (2002). Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data*, 8(2/3):257–274.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*, July.

Over, P., Dang, H., and Harman, D. (2007). DUC in context. *Inf. Process. Manage.*, 43:1506–1520, November.

Owczarzak, K. and Dang, H. (2010). Overview of the tac 2010 summarization track. In *Proceedings of TAC 2010*. NIST.

Paice, C. D. (1990). Constructing Literature Abstracts by Computer: Technics and Prospects. *Information Processing & Management*, 26(1):171–186.

Plaza, L., Díaz, A., and Gervás, P. (2008). Concept-graph based biomedical automatic summarization using ontologies. In *Proceedings of the 3rd Textgraphs Workshop on Graph-Based Algorithms for Natural Language Processing*, pages 53–56, Stroudsburg, PA, USA. Association for Computational Linguistics.

Plaza, L. (2011). *Towards a methodology for automatic term recognitionThe Use of Semantic Graphs in Automatic Summarization: Comparative Case Studies in Biomedicine, Journalism and Tourism*. Ph.D. thesis, Universidad Complutense de Madrid, Spain.

Radev, D. R., Allison, T., Blair-Goldensohn, S., Blitzer, J., Çelebi, A., Dimitrov, S., Drábek, E., Hakim, A., Lam, W., Liu, D., Otterbacher, J., Qi, H., Saggion, H., Teufel, S., Topper, M., Winkel, A., and Zhang, Z. (2004). MEAD - A Platform for Multidocument Multilingual Text Summarization. In *LREC*.

Saggion, H. and Gaizauskas, R. (2004). Multi-document summarization by cluster/profile relevance and redundancy removal. In *Proceedings of the Document Understanding Conference 2004*. NIST.

Saggion, H. and Poibeau, T. (2013). Automatic text summarization: Past, present, and future. In Poibeau, T., Saggion, H., Piskorski, J., and Yangarber, R., editors, *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing. Springer.

Saggion, H., Bontcheva, K., and Cunningham, H. (2003). Robust Generic and Query-based Summarisation. In *Proceedings of the European Chapter of Computational Linguistics (EACL), Research Notes and Demos*.

Saggion, H., Moreno, J. M. T., da Cunha, I., SanJuan, E., and Velzquez-Morales, P. (2010). Multilingual summarization evaluation without human models. In Huang, C.-R. and Jurafsky, D., editors, *COLING*, pages 1059–1067. Chinese Information Processing Society of China.

Saggion, H. (2008). SUMMA: A Robust and Adaptable Summarization Tool. *Traitement Automatique des Langues*, 49(2):103–125.

Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

Tombros, A., Sanderson, M., and Gray, P. (1998). Advantages of Query Biased Summaries in Information Retrieval. In *Intelligent Text Summarization. Papers from the 1998 AAAI Spring Symposium. Technical Report SS-98-06*, pages 34–43, Standford (CA), USA, March 23-25. The AAAI Press.