Language-Independent Multi-Document Text Summarization with Document-Specific Word Associations

Oskar Gross
Department of Computer Science and HIIT
University of Helsinki, Finland
oskar.gross@cs.helsinki.fi

Antoine Doucet
Laboratoire Informatique, Image et Interaction
University of La Rochelle, France
antoine.doucet@univ-lr.fr

Hannu Toivonen
Department of Computer Science and HIIT
University of Helsinki, Finland
hannu.toivonen@cs.helsinki.fi

ABSTRACT

The goal of automatic text summarization is to generate an abstract of a document or a set of documents. In this paper we propose a word association based method for generating summaries in a variety of languages. We show that a robust statistical method for finding associations which are specific to the given document(s) is applicable to many languages. We introduce strategies that utilize the discovered associations to effectively select sentences from the document(s) to constitute the summary. Empirical results indicate that the method works reliably in a relatively large set of languages and outperforms methods reported in MultiLing 2013.

CCS Concepts

•**Information systems** → **Summarization;** *Data mining;* •**Computing methodologies** → *Semantic networks;* Information extraction; •**Applied computing** → Digital libraries and archives;

Keywords

Natural language processing; text summarization; text mining; co-occurrence analysis

1. INTRODUCTION

The amount of information on the Internet is growing so rapidly that methods which are able to make it consumable for users, e.g., by summarization, are becoming more important every day. The problem is emphasized with news stories, where several news providers report on same events using similar facts. Automatic text summarization is one way to solve this problem by creating a comprehensive summary of a given set of documents. Effective summarization potentially makes it much easier for the readers to obtain the information efficiently.

In text summarization, one or more documents on some topic are abstracted into a shorter text. Summarization methods are needed essentially for all written languages but developing them separately is a huge effort. Motivated by this need, we introduce a summarization method that makes only some minimal assumptions about the language: that the text can be split to sentences (based on punctuation) and sentences further to words (based on white space). In the experiments of this paper, we applied it successfully on nine different languages without any language-specific resources, tools, or tuning.

The method we propose analyses co-occurrences of words in the given document and uses this information to pick suitable sentences from the document to produce a summary for it. It has been shown before that discovery of document-specific associations works well for summarization of the English language [12]. In this paper we extend the method and show that this method is more universal; in particular, we apply it to many languages and to multi-document summarization.

A central task in text summarization is to detect what is important in the given documents. The crux of the method proposed here is to statistically identify pairwise word associations which are characteristic and specific to these documents. To a degree this is similar to finding relevant words, e.g., using tf-idf. Obviously, associations (i.e., pairs of words) are more informative than individual words.

For instance, *accident* is a frequent word in news stories, and so is *Obama* at the time of writing of this paper. A hypothetical document talking about an accident to President Obama is characterized by the *combination* of these two common words, and our goal is to be able to recognize such unexpected combinations. In contrast, a purely keyword-based method fails to discover the connection, and may actually miss both words if they are sufficiently common in news in general.

On the other hand, the combination of *Obama* and *president* in a news story is not interesting since it is not unexpected. The method we use therefore down-weighs word pairs that are frequent in general.

The main contributions of the paper are the following.

- Using document-specific word associations as a model of the document, we propose two novel measures of how well a summary represents that model. Both outperform the previous measure based on word associations [12]. We also consider two alternative optimization techniques to find a set of sentences to be used as the summary; one technique is a greedy one, the other one uses a genetic algorithm.

SAC 2016, April 04 - 08, 2016, Pisa, Italy

ISBN 978-1-4503-3739-7/16/04...$15.00

DOI: http://dx.doi.org/10.1145/2851613.2851647

- The method is based on a simple model: a document is a set of sentences, and each sentence is a set of words. This makes the method easily applicable to different languages; we have used it on nine languages without any language-specific pre-processing at all. The model also makes multi-document summarization trivial: a set of documents is simply a larger set of sentences.
- The method outperforms existing methods when tested in multi-document summarization tasks in nine different languages; we evaluated the method experimentally on the tasks of MultiLing 2013 [9], an event for multilingual multi-document summarization. In six languages it gives the best results, in the remaining three it is among the best ones.

The strong empirical results in document summarization suggest that document-specific associations do capture essential aspects of the documents across several languages. There probably are other applications for such automatically extracted information besides summarization.

In the rest of the paper we will first give an overview of related work in language-independent text summarization. In Section 3 we describe the problem formally. We continue by introducing the method in Section 4. The performance of the method is assessed empirically in Section 5. The paper is concluded in Section 6.

2. RELATED WORK

Text summarization is the task of automatically building short summaries of longer documents. It is a well-studied area, addressed with two main approaches. The first approach is to select existing sentences (or phrases or words) to form the summaries, in what is termed "extraction-based" summarization. In contrast, "abstraction-based" methods use natural language generation methods to represent the original document in a condensed form. Hybrids exist where sentences are altered, using techniques such as sentence compression around the key parts of the text. In addition, all of these approaches can either be supervised or unsupervised.

In this paper, we focus on unsupervised approaches, in which there is no human intervention in the summarization process whatsoever. An exhaustive review of such techniques is provided by Nenkova and McKeown [20]. Further, we focus on extraction-based approaches.

To perform unsupervised summarization, several techniques rely on Latent Semantic Analysis (LSA) [5] as their basis (e.g. [11]). An example of purely unsupervised summarization is the DSDR method of He et al. [13]. This approach generates a summary by using sentences that best "reconstruct" the original document, in its diversity. This work has been extended by combining document reconstruction and topic decomposition [23].

An approach more closely related to ours is that of Baralis et al. [1, 2] who treat sentences as sets of items (i.e., words) and choose the sentences by using an approach based on frequent weighted itemsets. The difference to our method is that we neither use frequent itemsets nor association rules but exploit all pairs of words co-occurring in the same sentence. Perhaps more importantly, our method calculates its measure of relevance of associations by incorporating information from a background corpus, in order to contrast the document against general expectations about word associations.

A number of unsupervised methods take advantage of additional linguistic resources. In particular, the Two-Tiered Topic model by Celikyilmaz [4] uses Wordnet [19] and the DUC-provided user query for selecting the summary sentences. The Document Understanding Conference(DUC) provides most evaluation procedures and collections in the summarization field.

In this paper, our applications are in the specific task of multi-document summarization, in which a single summary needs to be constructed for a set of documents written about the same topic. This task has been shown to be more complex than single-document summarization as a larger set of documents inevitably induces a wider thematic diversity [17, 18].

Few techniques are language-independent, unsupervised and effective also in multi-document summarization. The most successful approach of the multilingual multi-document summarization workshop (MultiLing 2013) was UWB [22], a method based on singular value decomposition (SVD). UWB performed best in almost all the languages tested in MultiLing 2013.

3. PROBLEM

We will next formulate the problem of text summarization. Since the evaluation of summaries is an integral part of the problem, we also discuss methods to evaluate the generated summaries.

Formulation.

Let U be the universe of all possible sentences. We denote by D the given set of documents to be summarized. We ignore sentence order in documents, so each document $d \in D$ is simply a subset of all possible sentences, $d \subset U$.

Given a set of documents D, consisting in total of c words, the task is to summarize it into a document $\hat{d}$ consisting of at most k words, where $k \ll c$. Conceptually the goal is to create a document $\hat{d}$ such that the information contained in $\hat{d}$ is in some sense as similar to the document set D as possible:

$$\hat{d} = \max_{\substack{d' \subset U; \\ ||d'|| \leq k}} sim(d', D),$$

where d' can be any set of sentences consisting of at most k words.

In extraction-based summarization, the universe U is restricted to the sentences found in D, i.e., $U = \bigcup_{d \in D} d$.

Evaluation.

Evaluation of summaries is difficult since the similarity function $sim()$ above is difficult if not impossible to define objectively. In practical evaluations of summaries, it usually is based on human assessment, or on some rough computational similarity measure between a computer-generated summary and human-written summaries.

A classical method for automatic evaluation of summaries is ROUGE [15], also used in this paper. ROUGE uses n-gram analysis to calculate a similarity between human-written model summaries and automatically generated summaries. According to Lin and Hovy [16], ROUGE-1 score corresponds best to human judgement. Giannakopoulos and Karkaletsis have also proposed graph based measures AutoSummENG and MeMog for evaluating summaries [10].

They show that these measures correlate well with the ROUGE-2 measure.

Complexity.

Extraction-based summarization is a restricted version of the general summarization problem. Under some reasonable assumptions the problem then reduces to a (weighted) set cover problem [14]: we have to choose a set $\hat{d} \subset U$ of sentences such that $\hat{d}$ maximally covers the information in D.

The set cover problem is known to be NP-hard, so even if $sim()$ could be defined optimally and even if it could be computed efficiently, the problem would still remain computationally hard.

4. METHOD

The key problem in extraction-based summarization is how to measure the importance of a sentence. We use a method that estimates the importance of word pairs in the given document, and then weighs a sentence by the word pairs it contains.

4.1 Defining Document-Specific Word Associations

Let us start with some notation and simplifying assumptions we make.

Let T denote the set of all words. A sentence s is, in our model, simply a set $s = \{t_1, \cdots, t_k\}$ of words $t_i \in T$, i.e., we ignore the order of words.

In the case of multi-document summarization, as in the experiments of this paper, we simply consider the documents to be summarized as one long document $d_s = \bigcup_{d \in D} d$. Multi-document summarization is thus trivially reduced to single-document summarization.

Mixture Model for Word Co-occurrence.

The goal is to identify word associations that are more common in the document than expected. We next describe the statistical Mixture model for what is considered "expected", following Gross et al. [12]. Co-occurrences of words are here considered on sentence level. The Mixture model considers and combines two aspects of what is expected.

First, if t_i = *Obama* and t_j = *Putin* are both frequent within a document, then it is likely that they also co-occur several times in the same sentence within the document. To estimate this probability, the method needs frequencies of t_i and t_j in the document d_s to be summarized. These are denoted by n_i and n_j, respectively, while n_{ij} denotes the frequency of their co-occurrence and n the total number of sentences in d_s. Assuming that t_i and t_j are statistically independent, their expected frequency of co-occurrence is then $E_d(n_{ij}) = n_i \cdot n_j / n$.

Second, if the pair t_i = *Barack* and t_j = *USA* co-occurs frequently in news stories in general, then their co-occurrence is not unexpected in a given news document d_s. In order to estimate how often word pairs are likely to co-occur in general, the method also computes word and word pair frequencies in a background corpus $\mathcal{B}$. These frequencies are denoted by m_i, m_j, m_{ij}, and m, similarly to the counts obtained for the document d_s. The expected frequency of co-occurrence in d_s then is $E_{\mathcal{B}}(n_{ij}) = n \cdot m_{ij}/m$.

The Mixture model combines these two models and estimates the probabilities of words t_i and t_j and of their co-occurrence, denoted by p_i, p_j and p_{ij}, respectively, as

$$\begin{aligned} p_i &= (n_i + m_i)/(n + m), \\ p_j &= (n_j + m_j)/(n + m), \\ p_{ij} &= (n_i \cdot n_j/n + m_{ij})/(n + m). \end{aligned}$$

Probabilities p_i and p_j are obtained in a straightforward manner from the frequencies of t_i and t_j, respectively, in the union of $\mathcal{B}$ and d_s. This definition of p_i equals the average of probabilities n_i/n and m_i/m weighted by their sample sizes n and m, respectively (and similarly for p_j).

The probability p_{ij} of co-occurrence is conceptually also estimated from the union of $\mathcal{B}$ and d_s, but *not* using the observed frequency of co-occurrence n_{ij} in d_s—since we want to estimate if it is unexpected or not—but instead under the assumption that words t_i and t_j are statistically independent in d_s. This definition equals the average of probabilities $E_d(n_{ij})/n$ and $E_{\mathcal{B}}(n_{ij})/n$, weighted again by the sample sizes n and m, respectively.

As can be seen from above, the method combines two models into one mixture model: one based on the document itself, another one based on the background; hence the name *Mixture*. The motivation for using the Mixture model is that we cannot always assume that the distributions between the background and the document are similar, thus we draw from both models.

Weighting Word Associations.

We use log-likelihood ratio (llr) to measure the unexpectedness of word associations in d_s [6]. The measure compares the fit of two multinomial models to the data, one is a null model and the other is an alternative model. The null model is the Mixture model described above, defining what is expectable under the assumptions of the model. The alternative model is the maximum likelihood model obtained from d_s, where probabilities are estimated directly from the document itself: $q_i = n_i/n$; $q_j = n_j/n$; $q_{ij} = n_{ij}/n$.

The multinomial models actually have as their parameters the probabilities of the mutually exclusive cases of co-occurrence of t_i and t_j (p_{ij}; already known from above), of occurrence of t_i without t_j (denoted by p_{i-j}), of occurrence of t_j without t_i (denoted by p_{-ij}), and of absence of both (denoted by p_{-i-j}). We can obtain these parameters easily from the previously defined probabilities: $p_{i-j} = p_i - p_{ij}$; $p_{-ij} = p_j - p_{ij}$; $p_{-i-j} = 1 - p_{ij} - p_{i-j} - p_{-ij}$.

The log-likelihood ratio is then computed as [6, 12]

$$LLR(t_i, t_j) = 2 \sum_{\substack{a \in \{\text{“}ij\text{”}, \text{“}i-j\text{”}, \\ \text{“}-ij\text{”}, \text{“}-i-j\text{”}\}}} n_a (\log p_a - \log q_a).$$

Document-specific associations are now obtained by selecting those word pairs for which the log-likelihood ratio is greater than zero, $LLR(t_i, t_j) > 0$, and which co-occur at least twice in the document d_s. The latter condition reduces noise caused by rare words and co-occurrences.

4.2 Sentence Selection

Document-specific associations presumably carry essential information about the document, and earlier results indicate that this is indeed the case, at least in English [12]. The next task is to take advantage of the discovered document-specific associations and pick sentences from the document to generate a summary of it. In this paper, we will use three

strategies: a) pick sentences that cover as many of the associations as possible [12]; b) pick sentences that cover the most central nodes in the term-association graph; c) combine the two strategies above. We will next define sentence-scoring functions for these three strategies, and then will consider two optimization methods for picking the best possible sentences.

As some of the components also incorporate graph algorithms, we also consider a graph $G = (V, E, W)$, where $V = \bigcup_{s \in d_s} s$ is the set of nodes (all words in the document d_s),

$$E = \{\{t_i, t_j\} \mid t_i \neq t_j, \exists s \in d_s \text{ s.t. } \{t_i, t_j\} \subset s\}$$

is the set of edges (associations between words), and $W : V \times V \rightarrow \mathcal{R}$ maps an edge e to a positive real number (i.e. edge weight). The log-likelihood ratio LLR is used as the edge weight, i.e., $W(t_i, t_j) = LLR(t_i, t_j)$.

Covering Associations.

The assumption given above is that stronger associations cover the most important relations between words in the given document. Hence, having as many of the most important associations also in the generated summary is a natural goal [12]. However, rather than aiming to actually replicate the document-specific associations in the summary, the aim is to have many of them as word co-occurrences. In other words, the goal is to pick sentences so that the words of each important association co-occur in at least one sentence of the summary. This choice is motivated by the need to produce short summaries; statistics based on the number of co-occurrences would indeed have large variance and would not likely be reliable.

This task now reduces to the weighted set cover problem. The best summary consists of the set of sentences that covers as many of the heaviest associations (edges) as possible. The score of a summary S (a set of sentences) is

$$cover(S) = \sum_{\substack{e \in E: \\ \exists s \in S \text{ s.t. } e \subset s}} W(e),$$

and the summarization task now reduces to finding the set S of sentences that matheximizes the score, when the size of the summary S is constrained to at most k words.

Covering Central Words.

We propose word-centrality as an alternative measure to the graph coverage above. We still use word-associations, but instead of covering associations (edges in the word association graph), we aim to cover important words (nodes in the word association graph). The rationale here is that the most central words in the graph induced by pairwise associations are central concepts of the document.

To measure the importance of words, given word associations, we use the document graph G and calculate the closeness centrality [8] for each of the nodes in the graph. For a node $v \in V$, the centrality is

$$C(v) = \frac{|V|}{\sum_{u \in V} d(u, v)},$$

where $d(u, v)$ is the length of the shortest path between nodes u and v; the length of a path is computed as the sum of inverse weights $1/W(e)$ of its edges.

Similarly to the association cover, we now obtain a centrality score for summary S as follows:

$$centrality(S) = \sum_{\substack{v \in V: \\ \exists s \in S \text{ s.t. } v \in s}} C(v).$$

Covering Associations and Central Words.

While both measures above are based on document-specific word associations, it possible that they capture different nuances of the document. In case these differences are complementary, some combination of the measures potentially outperforms either one.

We propose to define such a combination simply as their sum. However, to give both components roughly equal weight, we first normalize both scores to be between 0 and 1:

$$combined(S) = \frac{cover(S)}{\sum_{\{t_i, t_j\} \in E} W(t_i, t_j)} + \frac{centrality(S)}{\sum_{v \in V} C(v)}.$$

Greedy Optimization Strategy.

As was already noted above, the problem of selecting an optimal set of sentences to form summary S is NP-hard. We will use two alternative heuristics this: a greedy strategy is described in this subsection, and a method based a genetic algorithm in the next one.

The standard greedy algorithm first takes the sentence which covers as many word associations as possible, and then chooses the next sentence ignoring the already covered pairs [12]. We can directly apply the same greedy strategy also to cover central words, or the combined measure.

For the sake of simplicity, consider the graph G induced from the document d_s and an initially empty set $S = \emptyset$, to which sentences will be added to constitute the final summary. The score, according to which individual sentences s are selected by the greedy approach, normalizes the additional coverage given by a sentence s with its length $|s|$: $cover(s) = \sum_{e \in E:\ e \subset s} W(e)/|s|$.

Similarly, we obtain a sentence scoring function based on word centrality: $centrality(s) = \sum_{t \in s} C(t)/|s|$.

Algorithm 1 describes the greedy process for the original graph cover. It can be easily adapted for *centrality*() and *combined*(). The algorithm first selects the best-scoring sentence $\hat{s}$ and adds this to the summary, $S = \hat{s} \cup S$. The graph is next updated by removing from G all the edges between nodes which co-occur in $\hat{s}$. This step downweighs sentences that contain already covered pairs, and effectively also prevents selection of duplicates. The sentence selection and graph update process is then repeated until no more sentences can be added to the summary within the limit of k words. When applied to *centrality*() or *combined*(), a record must be kept of words not yet covered. However, node-centrality scores should *not* be updated in the process.

Genetic Algorithm.

The second approach for finding sentences which best cover the document-specific associations or words is an evolutionary algorithm. We chose the evolutionary algorithm as an alternative optimization method as it makes few as-

Algorithm 1 Greedy Selection Algorithm

```
procedure GreedySelect
    Input: d_s, a set of sentences to be summarized
    Output: S ⊂ d_s, a summary of d_s
    S ← ∅                                  ▷ An initially empty summary
    ls ← 0                                 ▷ Current summary length
    while ls < k do
        ŝ ← null
        ŝ ← argmax_{s∈d_s : |s|+ls≤k} cover(s)
        if ŝ = null then
            break
        end if
        S ← S ∪ ŝ
        for (t_i, t_j) ⊂ ŝ do
            W(t_i, t_j) ← 0
        end for
        ls ← ls + |s|
    end while
    return S
end procedure
```

Parameter	Value
λ, population size	300
μ, number of best individuals selected for producing offspring	100
crossover rate (probability of crossover)	0.3
mutation rate (probability of mutation)	0.7
number of iterations	150

Table 1: The parameters for the genetic algorithm.

sumptions about the underlying fitness landscape and it is easy to apply to different kinds of problems.

For the evolutionary algorithm we need to define the genome, mutation, crossover and scoring function. We defined the *genome* to be a set of sentences (technically, sentence identifiers). The *crossover* function is defined between two individuals a and b (sets of sentences) and produces two individuals a' and b' to the offspring. For each sentence in a and b we will uniformly randomly assign the sentence to a' or b'. The *mutation* function takes an individual a as input and generates a new modified individual a', by randomly adding or removing a random sentence in a'. The *scoring* function is either $cover(S)$, $centrality(S)$ or $combined(S)$. However, if the individual contains more than k words then the score is 0.

We used the $(\mu + \lambda)$ elitist strategy [21] for the optimization and the DEAP [7] package for its implementation.

In order to use the strategy there is a number of parameters to set. The parameter values were obtained by rough experimental analysis of the convergence speeds on the English language (Table 1).

5. EVALUATION

We next carry out an empirical evaluation of the proposed method. The general aims are to obtain a view to the overall performance of the method, and to the effects of its various components (scoring methods, optimization techniques). Specifically, we will look for answers to the following questions. (1) Which graph based scoring method captures the information of the documents best (i.e. *cover*(), *centrality*(), *combined*())? (2) What is the effect of the optimization strategy on the results (greedy vs. genetic algorithm)? (3) How reliably and consistently does the method work for different languages? (4) How does the method perform in comparison to other systems?

5.1 Experimental Setup

Evaluation Method.

We will use the ROUGE [15] evaluation method for evaluating summaries. The ROUGE method uses the overlap of n-grams between model summaries, written by humans, and generated summaries to measure the similarity. For instance, ROUGE-1 score just looks at unigrams, ROUGE-2 score looks at 2-grams and ROUGE-L looks for the longest common sequence between two texts. The ROUGE score breaks down into two components, precision and recall. For evaluation we will use the combined score, F-measure, computed as the harmonic mean between precision and recall. We originally attempted to use the evaluation method MeMoG [10], also used in MultiLing 2013, but were not able to reproduce the evaluation results published in MultiLing so we resorted to ROUGE standard instead.

Dataset.

We use the MultiLing-2013 [9] dataset to evaluate our method. The dataset contains documents in 10 different languages – English, French, Chinese, Romanian, Spanish, Hindi, Arabic, Hebrew, Greek and Czech. Our method assumes that the text has been (or can trivially be) broken to words. Since this assumption does not hold for Chinese, we omitted it from our experiments.

MultiLing contains 15 topics for each language except for French and Hindi, for which the number of topics is 10. On average each topic consists of 10 documents which need to be collectively summarized into a text of 250 words. In our case, this multi-document summarization task is trivially reduced to single-document summarization by concatenating the documents into one set of sentences. The background corpus consists of the documents in the same language except the documents being summarized.

Additionally, MultiLing 2013 has made available the summaries generated by systems that participated in the event. In our comparisons below with other systems, we have computed ROUGE scores etc. for the other systems from the original summaries they have provided, i.e., we did not reimplement nor re-run any of the systems.

Notation.

We have above proposed several alternative configurations for word-association based summarization, and we use the following notation to denote these configurations. First, the options for sentence-scoring are the graph cover measure (denoted by G), word centrality measure (C), and their combination ($G+C$). Second, optimization strategies for sentence selection are the greedy method (GR) and the genetic algorithm (GA). We refer to a combination of a scoring measure and an optimization method by their concatenation, e.g., G_GA refers to the graph cover scoring, optimized using a genetic algorithm.

5.2 Sentence Scoring Methods

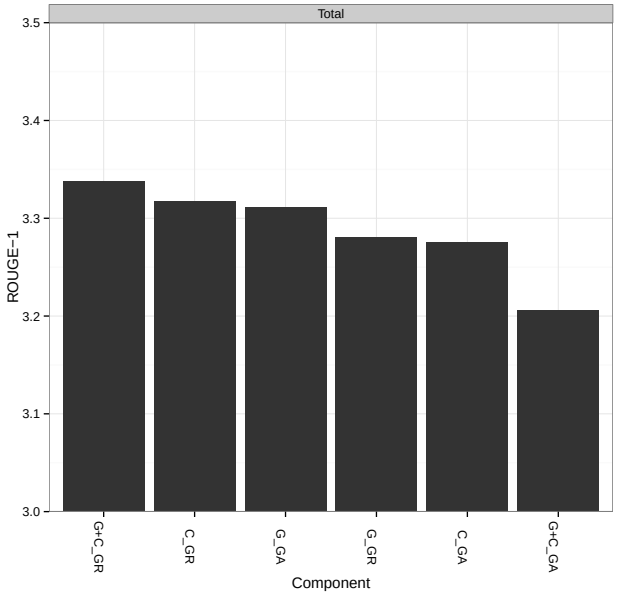

Figure 1: The performance of the different scoring methods with different optimization strategies. Note that the y-axis is limited to the range $3 - 3.5$.

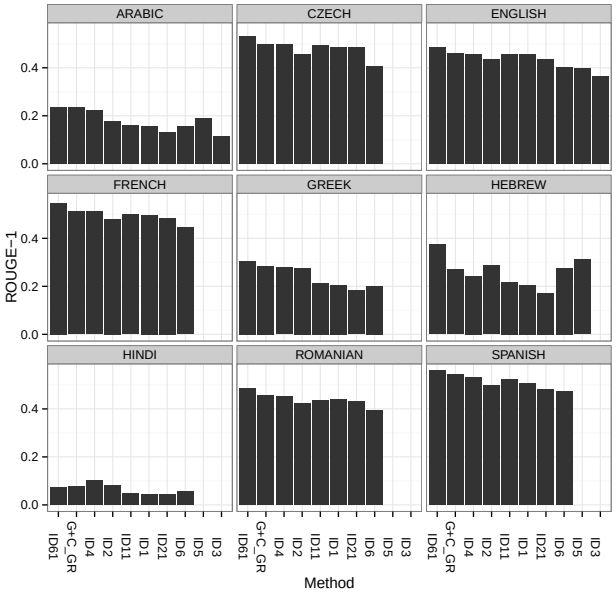

Figure 2: A comparison between all systems for all languages. Note that ID61 is not a real summarization method (see text).

First, we will take a look at how the different sentence scoring measures perform with different optimization methods. Instead of looking at individual languages here, we will compare the total scores obtained over all the languages with ROUGE-1. The scores for different combinations of scores and optimization strategies can be seen in Figure 1.

Best results are obtained with the combined measure $G + C$, followed by the word centrality-based measure C and then the graph cover based measure G. The differences between these measures are relatively small, however. For the question (1) we conclude that most likely both the word centrality measure and the graph association measure cover important parts of the document. As the $G + C$ measure performed in our experiment a bit better than either of the individual measures alone, it suggests that the measures do capture different nuances of the documents.

Between the two optimization methods (question 2), the greedy algorithm tends to perform better than the genetic algorithm (Figure 1). This is a slight surprise since the genetic algorithm should be able to explore a much wider space of possible summaries. The relatively poor performance of the genetic algorithm here is probably due to the simplistic setup; genetic algorithms designed specifically for the weighted set cover problem are known to produce better results than standard solutions [3]. The greedy method is known to be suboptimal, but a positive interpretation of the results here is that the greedy method actually performs well and cannot be easily outperformed.

5.3 Language-Wise Performance

Next we will take a look at summarization performances for individual languages, for the best variant $G + C\_GR$ of our method, as well as the participants of MultiLing 2013. Our main aim in this subsection is to compare the stability of the performance or our method in different languages; a systematic comparison to the other methods is provided in the next subsection.

Before going to the results, let us introduce the baseline methods for MultiLing 2013: a global baseline (ID6) and a global topline (ID61). The *global baseline* system ID6 is a simple vector space model based approach. It finds the centroid C in the vector space and tries to generate text which is most similar to the centroid, according to the cosine measure. The *global topline* method ID61 is not a real summarization method, it is an approximation of the upper limit of performance in extraction-based summarization. It works similarly to ID6, but "cheats" by using human-written summaries to generate the vector space, and then chooses sentences from the original documents to create text which is most similar to the centroid. Among the summarization methods of MultiLing 2013, ID4 denotes the best performing method, UWB [22]. For other methods we refer to the MultiLing 2013 overview paper [9].

Results over all methods and languages can be seen in Figure 2. There are two main observations to be made.

First, the proposed method is highly competitive against the other systems. It actually performs best among the automatic systems for six out of the nine languages (recall that ID61 is not an actual summarization system but an approximation of the upper limit). The method proposed here is outperformed only on Hebrew, Hindi and Czech.

Second, the results indicate that the proposed method is robust with regard to different languages, in the sense that it consistently ranks among the best ones and never loses much to the best one. On the other hand, some languages seem much more difficult for all methods, especially Hindi and Arabic, but also Greek and Hebrew, so robustness here does not mean equally good absolute performance over all

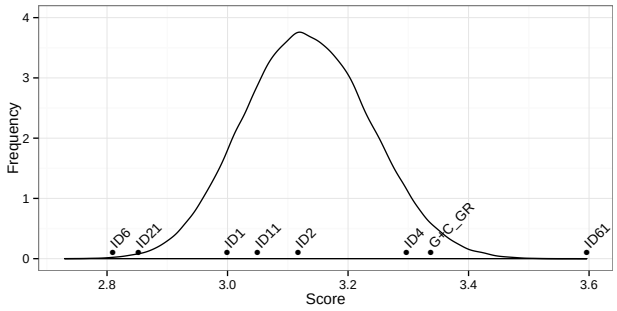

Figure 3: The permutation test for MultiLing 2013 systems comparison to G+C_GR method. Note that ID61 is not a real summarization method.

languages.

The answer to question (3) thus is that the proposed method seems to be generally applicable to many languages, with varying absolute performance but consistent relative performance in comparison to other methods applicable over a set of languages.

5.4 Statistical Comparison to Other Methods

Figure 2 already indicated strong relative performance of the method in comparison to other methods. We will now compare the performances of different methods statistically. We compare the total scores of our method, over all languages, to the scores of those methods that have results for all the languages in MultiLing 2013 (ID3 and ID5 were omitted since they only have results for some languages).

To avoid parametric assumptions about the distribution of scores, we carried out a permutation test as follows. The null hypothesis is that the proposed method is not statistically different from the other methods. In particular, for any given language, the proposed method could have received any of the scores that any method obtained for that language. Sampling a single random total score from this null hypothesis is easy: pick a random score for each language (among the ones obtained by the other systems) and sum up the scores.

By repeating this process 100 000 times we obtain an approximation of the distribution of total scores under the null hypothesis; this is shown as the curve in Figure 3. The total score of 3.337 obtained by our method can now be contrasted against the null distribution. The tail of the distribution starting from score 3.337 contains only 2.7% of the randomizations, i.e., the one-tailed empirical p-value is 0.027. Obviously, the same procedure can be used to obtain p-values for any of the methods.

Figure 3 also shows the total scores obtained by different methods. We can see that the global topline ID61 performs much better than any of the automatic systems. Among the real systems, the proposed method $G + C\_GR$ performs best, and is statistically significantly different from the other systems at level < 0.05 (empirical p-value 0.027). The significance level of ID4 is < 0.1 (empirical p-value 0.060).

A pairwise comparison between ID4 (UWB [22]) and $G + C\_GR$ using paired Wilcoxon rank sum test indicates that the methods are not statistically significantly different

Method	ROUGE-1	ROUGE-2	ROUGE-L
ID61	3.60	1.51	3.09
G+C_GR	**3.34**	1.28	**2.89**
ID4	3.30	**1.36**	2.87
ID2	3.12	1.06	2.70
ID11	3.05	1.13	2.61
ID1	3.00	1.10	2.58
ID21	2.85	1.01	2.44
ID6	2.81	0.86	2.25

Table 2: The average ROUGE scores for all the MultiLing 2013 methods. Note that ID61 is not a real summarization method (see text).

(p-value 0.20). Among the different configurations of our method we tested (Figure 1), ID4 would rank in the middle. On the other hand, even the worst of the configurations, the poorly optimized version $G + C\_GA$ of the same combined model, clearly outperforms the next best method, ID2.

Finally, Table 2 shows results also for ROUGE-2 and ROUGE-L. With ROUGE-2, ID4 (UWB) performs best, followed by the Mixture model. With ROUGE-L, the Mixture model wins again, with a small margin over ID4.

The answer to question (4) is that the performance of the proposed method is statistically significantly better than the performance of the other methods in general. It is not statistically significantly better than the UWB system [22] but the proposed method is more easily applicable to different languages: while UWB uses language-specific stop-word lists and various tunable parameters, the Mixture model has no parameters and uses no language specific resources except for a background corpus.

6. CONCLUSIONS

We have introduced a new method for automatically creating summaries for documents. The method is statistical in nature, and is based on analysis of the document itself, as well as comparing it to other documents. Word associations that are characteristic and specific to the given document are recognized first, and then a summary is constructed by picking those sentences from the document that best cover information in the strongest associations. We proposed new measures for the coverage that outperformed the previous measure [12].

The method is essentially language-independent: it only uses punctuation and white space to identify sentences and words. In our experiments, we did *not* use stemming or lemmatization, stopword lists, or any other language-specific tools or resources. These could probably be used to produce better results, but our goal here was to develop techniques that are readily applicable to a wide range of languages.

We evaluated the proposed method empirically using multi-document summarization tasks in nine different languages from MultiLing 2013. Overall, the method outperformed all methods that participated MultiLing: it ranked first in six languages out of nine, and was among the best ones in the remaining three. A statistical analysis shows that it is significantly better than the other methods in general (but not significantly better in a pairwise test than UWB [22], the best method of MultiLing 2013).

The superior performance of the method is striking given

its extreme simplifications. Sentences are treated simply as sets of words, and documents as sets of sentences. The multi-document summarization problem is trivially reduced to single-document summarization by taking the union of all documents. The method was successfully applied to nine different languages without any changes between languages. The results indicate strongly that document-specific word associations do capture central information of documents across several languages.

While the results are relatively speaking good, the summarization problem is all but solved. The coherence and fluency of generated summaries is an issue especially for methods based on sentence selection, such as ours. Further work is needed in making summaries better in these respects. Furthermore, interesting results could be obtained with hybrid approaches combining together language generation techniques and sentence selection techniques based on document-specific associations.

Acknowledgements.

This work has been supported by the European Commission (FET grant 611733, ConCreTe) and the Academy of Finland (decision 276897, CLiC).

7. REFERENCES

[1] E. Baralis and L. Cagliero. Learning from summaries: supporting e-learning activities by means of document summarization. *Emerging Topics in Computing, IEEE Transactions on*, (99):1–12, 2015.

[2] E. Baralis, L. Cagliero, A. Fiori, and P. Garza. Mwi-sum: A multilingual summarizer based on frequent weighted itemsets. *ACM Transactions on Information Systems*, 34(1):5:1–5:35, 2015.

[3] J. Beasley and P. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94(2):392–404, 1996.

[4] A. Celikyilmaz and D. Hakkani-Tür. Discovery of topically coherent sentences for extractive summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 491–499, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[5] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science (JAIS)*, 41(6):391–407, 1990.

[6] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74, 1993.

[7] F.-A. Fortin, D. Rainville, M.-A. G. Gardner, M. Parizeau, C. Gagné, et al. DEAP: Evolutionary algorithms made easy. *The Journal of Machine Learning Research*, 13(1):2171–2175, 2012.

[8] L. C. Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1979.

[9] G. Giannakopoulos. Multi-document multilingual summarization and evaluation tracks in acl 2013 multiling workshop. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 20–28, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[10] G. Giannakopoulos and V. Karkaletsis. AutoSummENG and MeMoG in evaluating guided summaries. In *Proceedings of the Text Analysis Conference (TAC 2011)*, Gaithersburg, Maryland, USA, November.

[11] Y. Gong and X. Liu. Generic text summarization using relevance measure and latent semantic analysis. In *24th international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25, New Orleans, LA, USA, September 2001.

[12] O. Gross, A. Doucet, and H. Toivonen. Document summarization based on word associations. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval*, pages 1023–1026, Gold Cost, Australia, 2014. ACM.

[13] Z. He, C. Chen, J. Bu, C. Wang, L. Zhang, D. Cai, and X. He. Document Summarization Based on Data Reconstruction. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 620–626, Toronto, Ontario, Canada, July 2012.

[14] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, Dec. 1974.

[15] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, pages 25–26, Barcelona, Spain, July 2004.

[16] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the ACL*, NAACL, Edmonton, Canada, May–June 2003.

[17] H. Liu, H. Yu, and Z. Deng. Multi-document summarization based on two-level sparse representation model. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence,*, pages 196–202, Austin, Texas, USA, January 2015.

[18] K. McKeown and D. R. Radev. Generating summaries of multiple news articles. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 74–82, New York, NY, USA, 1995. ACM.

[19] G. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[20] A. Nenkova and K. McKeown. A survey of text summarization techniques. In *Mining Text Data*, pages 43–76. Springer, 2012.

[21] H.-P. Schwefel. *Numerical optimization of computer models*. John Wiley & Sons, Inc., 1981.

[22] J. Steinberger. The UWB summariser at multiling-2013. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 50–54, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[23] Z. Zhang, H. Li, et al. TopicDSDR: combining topic decomposition and data reconstruction for summarization. In *Web-Age Information Management*, pages 338–350. Springer, 2013.