

# AllSummarizer system at MultiLing 2015: Multilingual single and multi-document summarization

**Abdelkrime Aries      Djamel Eddine Zegour      Khaled Walid Hidouci**

École Nationale Supérieure d'Informatique (ESI, ex. INI)

Oued-Smar, Algiers, Algeria

{ab\_aries, d\_zegour, w\_hidouci}@esi.dz

## Abstract

In this paper, we evaluate our automatic text summarization system in multilingual context. We participated in both single document and multi-document summarization tasks of MultiLing 2015 workshop.

Our method involves clustering the document sentences into topics using a fuzzy clustering algorithm. Then each sentence is scored according to how well it covers the various topics. This is done using statistical features such as TF, sentence length, etc. Finally, the summary is constructed from the highest scoring sentences, while avoiding overlap between the summary sentences. This makes it language-independent, but we have to afford preprocessed data first (tokenization, stemming, etc.).

## 1 Introduction

A document summary can be regarded as domain-specific or general-purpose, using the specificity as classification criterion (Hovy and Lin, 1998). We can, also, look at this criterion from language angle: language-specific or language-independent summarization. Language-independent systems can handle more than one language. They can be partially language-independent, which means they use language-related resources, and therefore you can't add a new language so easily. Inversely, they can be fully language-independent.

Recently, multilingual summarization has received the attention of the summarization community, such as Text Analysis Conference (TAC). The TAC 2011 workshop included a task called "MultiLing task", which aims to evaluate language-independent summarization algorithms on a variety of languages (Giannakopoulos et al., 2011). In

the task's pilot, there were seven languages covering news texts: Arabic, Czech, English, French, Greek, Hebrew and Hindi, where each system has to participate for at least two languages. MultiLing 2013 workshop is a community-driven initiative for testing and promoting multilingual summarization methods. It aims to evaluate the application of (partially or fully) language-independent summarization algorithms on a variety of languages. There were three tasks: "Multi-document multilingual summarization" (Giannakopoulos, 2013), "Multilingual single document summarization" (Kubina et al., 2013) and "Multilingual summary evaluation". The multi-document task uses the 7 past languages along with three new languages: Chinese, Romanian and Spanish. The single document task introduces 40 languages.

This paper contains a description of our method (Aries et al., 2013) which uses sentences' clustering to define topics, and then trains on these topics to score each sentence. We will explain each task in the system (AllSummarizer), especially the preprocessing task which is language-dependent. Then, we will discuss how we fixed the summarization's hyper-parameters (threshold and features) for each language. The next section (Section 5) is reserved to discuss the experiments conducted in the MultiLing workshop. Finally, we will conclude by discussing possible improvements.

## 2 Related works

Clustering has been used for summarization in many systems, either using documents as units, sentences or words. The resulted clusters are used to extract the summary. Some systems use just the biggest cluster to score sentences and get the top ones. Others take from each cluster a representative sentence, in order to cover all topics. While there are systems, like ours, which score sentences according to all clusters.

“CIST” (Liu et al., 2011; Li et al., 2013) is a system which uses hierarchical Latent Dirichlet Allocation topic (hLDA) model to cluster sentences into sub-topics. A sub-topic containing more sentences is more important and therefore those containing just one or two sentences can be neglected. The sentences are scored using hLDA model combined with some traditional features. The system participated for multi-document summarization task, where all documents of the same topic are merged into a big text document.

Likewise, “UoEssex” (El-Haj et al., 2011) uses a clustering method (K-Means) to regroup similar sentences. The biggest cluster is used to extract the summary, while other clusters are ignored. Then, the sentences are scored using their cosine similarities to the cluster’s centroid. The use of the biggest cluster is justified by the assumption that a single cluster will give a coherent summary.

The scoring functions of these two systems are based on statistical features like frequencies of words, cosine similarity, etc. In the contrary, systems like those of Conroy et al. (2011) (“CLASSY”), Varma et al. (2011) (“SIEL\_IITH”), El-Haj and Rayson (2013), etc. are corpus-based summarizers, which can make it hard to introduce new languages. “CLASSY” uses naïve Bayes to estimate the probability that a term may be included in the summary. The classifier was trained on DUC 2005-2007 data. As for backgrounds of each language, Wikinews are used to compute Dunning G-statistic. “SIEL\_IITH” uses a probabilistic Hyperspace Analogue to Language model. Given a word, it estimates the probability of observing another word with it in a window of size  $K$ , using a sufficiently large corpus. El-Haj and Rayson (2013) calculate the log-likelihood of each word using a corpus of words frequencies and the multiLing’13 dataset. The score of each sentence is the sum of its words’ log-likelihoods.

In our method (Aries et al., 2013), we use a simple fuzzy clustering algorithm. We assume that a sentence can express many topics, and therefore it can belong to many clusters. Also, we believe that a summary must take in consideration other topics than the main one (the biggest cluster). To score sentences, we use a scoring function based on Naïve Bayes classification. It uses the clusters for training rather than a corpus, in order to avoid the problem of language dependency.

### 3 System overview

One of multilingual summarization’s problem is the lack of resources such as labeled corpus used for learning. Learning algorithms were used either to select the sentences that should be in the summary, or to estimate the features’ weights. Both cases need a training corpus given the language and the domain we want to adapt the summarizer to. To design a language-neutral summarization system, either we adapt a system for input languages (Partly language-neutral), or we design a system that can process any language (Fully language-neutral).

Our sentence extraction method can be applied to any language without any modifications, affording the pre-process step of the input language. To do this, we had to find a new method to train our system other than using a corpus (language and topic dependent). The idea was to find different topics in the input text using similarity between sentences. Then, we train the system using a scoring function based on Bayes classification algorithm and a set of features to find the probability of a feature given the topic. Finally, we calculate for each sentence a score that reflects how it can represent all the topics.

In our previous work (Aries et al., 2013), our system used only two features which have the same nature (TF: uni-grams and bi-grams). When we add new features, this can affect the final result (summary). Also, our clustering method lies on the clustering threshold which has to be estimated somehow. To handle multi-document summarization, we just fuse all documents in the same topic and consider them as one document. Figure 1 represents the general architecture of AllSummarizer<sup>1</sup>.

#### 3.1 Preprocessing

This is the language-dependent part, which can be found in many information retrieval (IR) works. In our system, we are interested in four preprocessing tasks:

- Normalizer: in this step, we can delete special characters. For Arabic, we can delete diacritics (Tashkiil) if we don’t need them in the process (which is our case).
- Segmenter: The segmenter defines two func-

<sup>1</sup><https://github.com/kariminf/AllSummarizer>

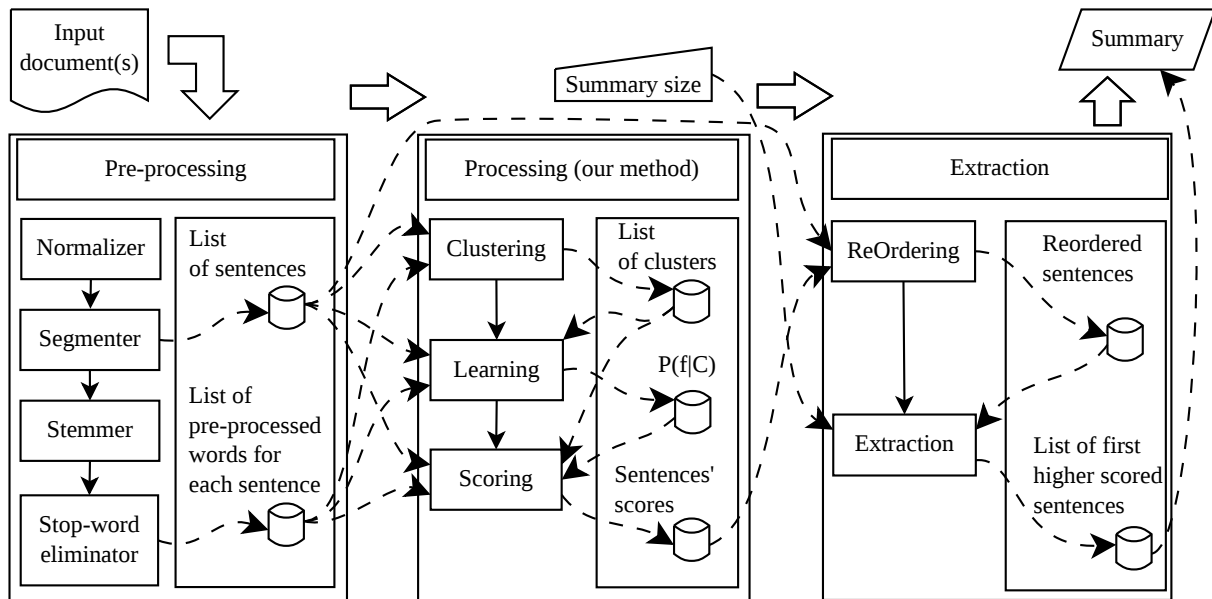


Figure 1: General architecture of AllSummarizer.

tions: sentence segmentation and word tokenization.

- **Stemmer:** The role of this task is to delete suffixes and prefixes so we can get the stem of a word.
- **Stop-Words eliminator:** It is used to remove the stop words, which are the words having no signification added to the text.

In this work, normalization is used just for Arabic and Persian to delete diacritics (Tashkiil). Concerning stop-word elimination, we use pre-compiled word-lists available on the web. Table 1 shows each language and the tools used in the remaining pre-processing tasks.

### 3.2 Topics clustering

Each text contains many topics, where a topic is a set of sentences having some sort of relationship between each other. In our case, this relationship is the cosine similarity between each two sentences. It means, the sentences that have many terms in common are considered in the same topic. Given two sentences  $X$  and  $Y$ , the cosine similar-

Table 1: Tools used to pre-process each language

Prerocess task	Tools	Languages
Sentence segmentation	openNLP <sup>2</sup>	Nl, En, De, It, Pt, Th
	JHazm <sup>3</sup>	Fa
	Regex	The remaining
Words tokenization	openNlp	Nl, En, De, It, Pt, Th
	Lucene <sup>4</sup>	Zh, Ja
	Regex	The remaining
	Shereen	Ar
Stemming	Khoja <sup>5</sup>	
	JHazm	Fa
	HebMorph <sup>6</sup>	He
	Lucene	Bg, Cs, El, Hi, Id, Ja, No
	Snowball <sup>7</sup>	Eu, Ca, Nl, En (Porter), Fi, Fr, De, Hu, It, Pt, Ro, Ru, Es, Sv, Tr
	/	The remaining

<sup>2</sup> <https://opennlp.apache.org/>

<sup>3</sup> <https://github.com/mojtaba-khallas/JHazm>

<sup>4</sup> <https://lucene.apache.org/>

<sup>5</sup> <http://zeus.cs.pacificu.edu/shereen/research.htm>

<sup>6</sup> <http://code972.com/hebmorph>

<sup>7</sup> <http://snowball.tartarus.org/>

ity between them is expressed by equation 1.

$$\cos(X, Y) = \frac{\sum_i x_i \cdot y_i}{\sqrt{\sum_i (x_i)^2} \cdot \sqrt{\sum_i (y_i)^2}} \quad (1)$$

Where  $x_i$  ( $y_i$ ) denotes frequencies for each term in the sentence  $X$  ( $Y$ ).

To generate topics, we use a simple algorithm (see algorithm 1) which uses cosine similarity and a clustering threshold  $th$  to cluster  $n$  sentences.

---

**Algorithm 1:** clustering method

---

```

Data: Pre-processed sentences
Result: clusters of sentences (C)
foreach sentence  $S_i / i = 1$  to  $n$  do
   $C_i += S_i$  ;
  //  $C_i$ :  $i$ th cluster
  foreach sentence  $S_j / j = i + 1$  to  $n$  do
     $Sim = \text{cosine similarity}(S_i, S_j)$  ;
    if  $sim > th$  then
      |  $C_i += S_j$  ;
    end
  end
   $C += C_i$  ;
end
foreach cluster  $C_i / i = n$  to  $1$  do
  foreach cluster  $C_j / j = i - 1$  to  $1$  do
    if  $C_i$  is included in  $C_j$  then
      |  $C -= C_i$  ;
      | break ;
    end
  end
end

```

---

### 3.3 Scoring function

A summary is a short text that is supposed to represent most information in the source text, and cover most of its topics. Therefore, we assume that a sentence  $s_i$  can be in the summary when it is most probable to represent all topics (clusters)  $c_j \in C$  using a set of features  $f_k \in F$ . We used Naïve Bayes, assuming independence between different classes and different features (a sentence can have multiple classes). So, the score of a sentence  $s_i$  is the product over classes of the product over features of its score in a specific class and feature (see equation. 2).

$$Score(s_i, \bigcap_j c_j, F) = \prod_j \prod_k Score(s_i, c_j, f_k) \quad (2)$$

The score of a sentence  $s_i$  in a specific class  $c_j$  and feature  $f_k$  is the sum of probability of the feature's observations when  $s_i \in c_j$  (see equation. 3). We add one to the sum, to avoid multiplying by a features' score of zero.

$$Score(s_i, c_j, f_k) = 1 + \sum_{\phi \in s_i} P(f_k = \phi | s_i \in c_j) \quad (3)$$

Where  $\phi$  is an observation of the feature  $f_k$  in the sentence  $s_i$ . For example, assuming the feature  $f_1$  is term frequency, and we have a sentence: “*I am studying at home.*”. The sentence after pre-processing would be:  $s_1 = \{\text{“stud” (stem of “study”), “home”}\}$ . So,  $\phi$  may be “stud” or “home”, or any other term. If we take another feature  $f_2$  which is sentence position, the observation  $\phi$  may take 1st, 2nd, 3rd, etc. as values.

### 3.4 Statistical features

We use 5 statistical features to score the sentences: unigram term frequency (TFU), bigram term frequency (TFB), sentence position (Pos) and sentence length (Rleng, PLeng).

Each feature divides the sentences to several categories. For example, if we have a text written just with three characters: a, b and c, and the feature is the characters of the text, then we will have three categories. Each category has a probability to occur in a cluster, which is the number of its appearance in this cluster divided by all cluster's terms, as shown in equation 4.

$$P_f(f = \phi | c_j) = \frac{|\phi \in c_j|}{\sum_{c_l \in C} |\phi' \in c_l|} \quad (4)$$

Where  $f$  is a given feature.  $\phi$  and  $\phi'$  are observations (categories) of the feature  $f$ .  $C$  is the set of clusters.

#### 3.4.1 Unigram term frequency

This feature is used to calculate the sentence pertinence depending on its terms. Each term is considered as a category.

#### 3.4.2 Bigram term frequency

This feature is similar to unigram term frequency, but instead of one term we use two consecutive terms.

#### 3.4.3 Sentence position

We want to use sentence positions in the original texts as a feature. The position feature used by Osborne (2002) divides the sentences into three

sets: the ones in the 8 first paragraphs, those in last 3 paragraphs and the others in between. Following the assumption that the first sentences and last ones are more important than the others.

Three categories of sentence positions seem very small to express the diversity between the clusters. Instead of just three categories, we divided the position space into 10 categories. So, if we have 20 sentences, we will have 2 sentences per category.

### 3.4.4 Sentence length

One other feature applied in our system is the sentence length (number of words), which is used originally to penalize the short sentences. Following a sentence's length, we can put it in one of three categories: sentences with length less than 6 words, those with length more than 20 words, and those with length in between Osborne (2002).

Like sentence position, three categories is a small number. Therefore, we used each length as a category. Suppose we have 4 sentences which the lengths are: 5, 6, 5 and 7, then we will have 3 categories of lengths: 5, 6 and 7.

In our work, we use two types of sentence length:

- Real length (RLeng): which is the length of the sentence without removing stop-words.
- Pre-processed length (PLeng): which is the length of the sentence after pre-processing.

### 3.5 Summary extraction

To extract sentences, we reorder them decreasingly using their scores. Then we extract the first non similar sentences until we get the wanted size (see algorithm 2).

## 4 Summarization parameters

In this section, we describe how the summarization parameters have been chosen.

The first parameter is the clustering threshold, which will lead to few huge clusters if it is small, and inversely. The clustering threshold is used with sentences' similarities to decide if two sentences are similar or not. Our idea is to use statistic measures over those similarities to estimate the clustering threshold. Eight measures have been used:

- The median

---

### Algorithm 2: extraction method

---

**Data:** input text

**Result:** a summary

add the first sentence to the summary;

**foreach** *sentence in the text* **do**

    calculate cosine similarity between this sentence and the last accepted one;

**if** *the similarity is under the threshold*

**then**

            | add this sentence to the summary;

**end**

**if** *the sum of the summary size and the current sentence's is above the maximum size* **then**

        | delete this sentence from the

        | summary;

**end**

**end**

---

- The mean
- The mode which can be divided to two: lower mode and higher mode, since we can have many modes.

- The variance

- $sDn = \frac{\sum |s|}{|D|*n}$

- $Dsn = \frac{|D|}{n*\sum |s|}$

- $Ds = \frac{|D|}{\sum |s|}$

Where,  $|s|$  is the number of different terms in a sentence  $s$ .  $|D|$  is the number of different terms in the document  $D$ .  $n$  is the number of sentences in this document.

The second parameter is the features' set, which is the combination of at least one of the five features described in section 3.4. We want to know which features are useful and which are not for a given language.

To fix the problem of the clustering threshold and the set of features, we used the training sets provided by the workshop organizers. For each document (or topic in multi-document), we generated summaries using the 8 measures of  $th$ , and different combinations of the scoring features. Then, we calculated the average ROUGE-2 score for each language. The threshold measure and the set of features that maximize this average will be used as parameters for the trained language.

Table 2 represents an example of the 10 languages and their parameters used for both tasks: MSS and MMS. We have to point out that the average is not always the best choice for the individual documents (or topic in multi-document). For example, in MSS, there is a document which gives a ROUGE-2 score of **0.28** when we use the parameters based on average scores. When we use the mean as threshold and just TFB as feature for the same document, we get a ROUGE-2 score of **0.31**.

## 5 Experiments

We participated in all workshop’s languages, either in single document or multi-document tasks. To compare our system to others participated systems, we followed these steps (for every evaluation metric):

- For each system, calculate the average scores of all used languages.
- For our system, calculate the average scores of used languages by others. For example, BGU-SCE-M team uses Arabic, English and Hebrew; We calculate the average of scores of these languages for this system and ours.
- Then, we calculate the relative improvement using the averages  $\frac{\text{our system} - \text{others system}}{\text{others system}}$ .

### 5.1 Evaluation metrics

In “Single document summarization” task, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) is used to evaluate the participated systems. It allows us to evaluate automatic text summaries against human made abstracts. The principle of this method is to compare N-grams of two summaries based on the number of matches between these two based on the recall measure. Five metrics are used: ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4 and ROUGE-SU4.

In “Multi-document summarization” task, Three metrics are officially used: AutoSummENG, MeMoG (Giannakopoulos and Karkaletsis, 2011) and NPower (Giannakopoulos and Karkaletsis, 2013).

### 5.2 Single document summarization

Besides our system (AllSummarizer), there are two more systems which participated in all 38 languages (EXB and CCS). Table 3 shows the comparison between our system and the other systems

in single document task, using the relative improvement.

Looking at these results, our system took the fifth place out of seven participants. It outperforms the Lead baseline. It took the last place out of three participants in all 38 languages.

### 5.3 Multi-document summarization

Besides our system (AllSummarizer), there are 4 systems that participated with all the 10 languages. Table 4 shows a comparison between our system and the other systems in multi-document task, using the relative improvement. We used the parameters fixed for single document summarization to see if the same parameters are applicable for both single and multi-document summarizations.

Looking to the results, our system took the seventh place out of ten participants. When we use single document parameters, we can see that it doesn’t outperform the results when using the parameters fixed for multi-document summarization. This shows that we can’t use the same parameters for both single and multi-document summarization.

## 6 Conclusion

Our intension is to create a method which is language and domain independent. So, we consider the input text as a set of topics, where a sentence can belong to many topics. We calculated how much a sentence can represent all the topics. Then, the score is used to reorder the sentences and extract the first non redundant ones.

We tested our system using the average score of all languages, in single and multi-document summarization. Compared to other systems, it affords fair results, but more improvements have to be done in the future. We have to point out that our system participated in all languages. Also, it is easy to add new languages when you can afford tokenization and stemming.

We fixed the parameters (threshold and features) based on the average score of ROUGE-2 of all training documents. Further investigations must be done to estimate these parameters for each document based on statistical criteria. We want to investigate the effect of the preprocessing step and the clustering methods on the resulted summaries. Finally, readability remains a challenge for extractive methods, especially when we want to use a multilingual method.

Table 2: Example of the parameters used for MSS and MMS.

Lang	Single document (MSS)		Multidocument (MMS)	
	Th	Features	Th	Features
Ar	Ds	TFB, Pos, PLeng	Ds	TFB, Pos, RLeng, PLeng
Cs	HMode	TFU, TFB, Pos, PLeng	Ds	TFB, Pos, PLeng
El	Median	TFU, TFB, Pos, RLeng, PLeng	LMode	TFB, RLeng
En	Median	TFU, Pos, RLeng, PLeng	LMode	TFB, Pos, RLeng, PLeng
Es	sDn	TFB, PLeng	Ds	TFB, PLeng
Fr	Median	TFB, Pos, RLeng	Mean	TFU, TFB, Pos, PLeng
He	Ds	TFB, PLeng	Median	TFB, RLeng, PLeng
Hi	/	/	Ds	TFB, Pos, RLeng, PLeng
Ro	HMode	TFB, RLeng, PLeng	sDn	TFB, Pos, PLeng
Zh	HMode	TFB, RLeng, PLeng	sDn	TFU, Pos, RLeng, PLeng

Table 3: Relative improvement of our method against other methods on the MultiLing 2015 Single document testing dataset

Methods	Our method improvement %				
	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-SU4
BGU-SCE-M (ar, en, he)	-09.19	-14.02	-19.39	-25.12	-11.07
EXB (all 38)	-07.64	-10.55	-09.86	-07.92	-10.63
CCS (all 38)	-07.33	-13.24	-10.95	-03.04	-07.40
BGU-SCE-P (ar, en, he)	-04.33	-01.63	-02.69	-06.16	-01.89
UA-DLSI (en, de, es)	+02.12	+06.25	+13.86	+17.15	+05.62
NTNU (en, zh)	+06.44	+07.06	+11.50	+21.81	+05.74
Oracles (all 38) [TopLine]	-31.64	-49.00	-63.80	-72.91	-36.77
Lead (all 38) [BaseLine]	+02.39	+08.67	+08.20	+04.02	+05.82

Table 4: Relative improvement of our method against other methods on the MultiLing 2015 multi-document testing dataset. *The minus sign means that the system participated in all languages except those mentioned.*

SysID	Our method improvement %		
	AutoSummENG	MeMoG	NPower
UJF-Grenoble (fr, en, el)	-08.87	-14.55	-03.62
UWB (all 10)	-22.56	-22.66	-07.54
ExB (all 10)	-09.44	-09.16	-02.80
IDA-OCCAMS (all 10)	-17.11	-17.68	-05.53
GiauUngVan (- zh, ro, es)	-16.43	-19.40	-05.68
SCE-Poly (ar, en, he)	-05.72	-03.35	-01.46
BUPT-CIST (all 10)	+10.67	+11.53	+02.85
BGU-MUSE (ar, en, he)	+05.67	+06.92	+01.74
NCSR/SCIFY-NewSumRerank (- zh)	+01.53	-01.25	+00.13
our system (MSS parameters) (all 10)	+01.98	+02.35	+00.58

## References

- Abdelkrime Aries, Houda Oufaida, and Omar Nouali. 2013. Using clustering and a modified classification algorithm for automatic text summarization. volume 8658 of *Proc. SPIE*, pages 865811–865811–9.
- John M. Conroy, Judith D. Schlesinger, and Jeff Kubina. 2011. Classy 2011 at TAC: Guided and multi-lingual summaries and evaluation metrics. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)–MultiLing Pilot Track.*, Gaithersburg, Maryland, USA.
- Mahmoud El-Haj and Paul Rayson. 2013. Using a keyness metric for single and multi document summarisation. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 64–71, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Mahmoud El-Haj, Udo Kruschwitz, and Chris Fox. 2011. University of Essex at the TAC 2011 multilingual summarisation pilot. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)–MultiLing Pilot Track.*, Gaithersburg, Maryland, USA.
- George Giannakopoulos and Vangelis Karkaletsis. 2011. Autosummeng and memog in evaluating guided summaries. In *TAC 2011 Workshop*, Gaithersburg, MD, USA. NIST.
- George Giannakopoulos and Vangelis Karkaletsis. 2013. Together we stand npower-ed. In *Proceedings of CICLing 2013*, Karlovasi, Samos, Greece.
- G. Giannakopoulos, M. El-Haj, B. Favre, M. Litvak, J. Steinberger, and V. Varma. 2011. Tac 2011 multiling pilot overview. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)–MultiLing Pilot Track.*, Gaithersburg, Maryland, USA.
- George Giannakopoulos. 2013. 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. Association for Computational Linguistics.
- Eduard Hovy and Chin-Yew Lin. 1998. Automated text summarization and the SUMMARIST system. In *Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998*, pages 197–214. Association for Computational Linguistics.
- Jeff Kubina, John Conroy, and Judith Schlesinger. 2013. Acl 2013 multiling pilot overview. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 29–38, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Lei Li, Corina Forascu, Mahmoud El-Haj, and George Giannakopoulos. 2013. Multi-document multilingual summarization corpus preparation, part 1: Arabic, english, greek, chinese, romanian. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 1–12, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10.
- Hongyan Liu, Pingan Liu, Wei Heng, and Lei Li. 2011. The cist summarization system at TAC 2011. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)–MultiLing Pilot Track.*, Gaithersburg, Maryland, USA.
- Miles Osborne. 2002. Using maximum entropy for sentence extraction. In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4, AS '02*, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Vasudeva Varma, Sudheer Kovelamudi, Jayant Gupta, Nikhil Priyatam, Arpit Sood, Harshit Jain, Aditya Mogadala, and Srikanth Reddy Vaddepally. 2011. IIT hyderabad in summarization and knowledge base population at TAC 2011. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)–MultiLing Pilot Track.*, Gaithersburg, Maryland, USA.