Three Statistical Summarizers at CLEF-INEX 2014 Tweet Contextualization Track

Juan-Manuel Torres-Moreno[1,2]

[1] École Polytechnique de Montréal - Département de Génie Informatique
CP 6079 Succ. Centre Ville H3C 3A7 Montréal (Québec), Canada.
[2] Université d'Avignon et des Pays de Vaucluse
BP 911228, 84911 Avignon Cedex 9, France
juan-manuel.torres@univ-avignon.fr
http://lia.univ-avignon.fr/chercheurs/torres/

Abstract. According to the organizers, the objective of the 2014 CLEF-INEX Tweet Contextualization Task is: "*...The Tweet Contextualization aims at providing automatically information - a summary that explains the tweet. This requires combining multiple types of processing from information retrieval to multi-document summarization including entity linking.*" We present three statistical summarizer systems applied to the CLEF-INEX 2014 task. Cortex summarizer uses several sentence selection metrics and an optimal decision module to score sentences from a document source. Artex summarizer uses a simple inner product among the topic-vector and the pseudo-word vector. Reg summarizer is a performant graph-based summarizer. The results show that our systems performed well on CLEF-INEX task. Our three systems have obtained the first rank in the INEX manual evaluation.

Keywords: INEX, Automatic Text Summarization, Tweet contextualization, Cortex, Artex.

1 Introduction

Automatic text summarization is indispensable to cope with ever increasing volumes of valuable information. An abstract is by far the most concrete and most recognized kind of text condensation [1, 2]. We adopted a simpler method, usually called *extraction*, that allows to generate summaries by extraction of relevant sentences [2–5]. Essentially, extracting aims at producing a shorter version of the text by selecting the most relevant sentences of the original text, which we juxtapose without any modification. The vector space model [6, 7] has been used in information extraction, information retrieval, question-answering, and it may also be used in text summarization [8]. CORTEX[3] is an automatic summarization system [9] which combines several statistical methods with an optimal decision algorithm, to choose the most relevant sentences.

[3] *CORTEX es Otro Resumidor de TEXtos* (CORTEX is anotheR TEXt summarizer).

An open domain Question-Answering system (QA) has to exactly answer a question expressed in natural language. QA systems are confronted with a fine and difficult task because they are expected to supply specific information and not whole documents. Currently there exists a strong demand for this kind of text processing systems on the Internet. A QA system comprises, *a priori*, the following stages [10]:

- Transform the questions into queries, then associate them to a set of documents;
- Filter and sort these documents to compute various degrees of similarity;
- Identify the sentences which might contain the answers, then extract text fragments from those that constitute the answers. In this phase an analysis using Named Entities (NE) is essential to find the expected answers.

Most research efforts in summarization emphasize generic summarization [11–13]. User query terms are commonly used in information retrieval tasks. However, there are few papers in literature that propose to employ this approach in summarization systems [14–16]. In the systems described in [14], a learning approach is used (performed). A document set is used to train a classifier that estimates the probability that a given sentence is included in the extract. In [15], several features (document title, location of a sentence in the document, cluster of significant words and occurrence of terms present in the query) are applied to score the sentences. In [16] learning and feature approaches are combined in a two-step system: a training system and a generator system. Score features include short length sentence, sentence position in the document, sentence position in the paragraph, and tf.idf metrics. Our generic summarization system includes a set of eleven independent metrics combined by a Decision Algorithm. Query-based summaries can be generated by our systems using a modification of the scoring method. In both cases, no training phase is necessary in our system.

This paper is organized as follows. In Section 2 we explain the CLEF-INEX 2014 Tweet Contextualization Track. In Section 3.1 we explain the methodology of our work. Experimental settings and results obtained with Cortex summarizer are presented in Section 5. Section 7 exposes the conclusions of the paper and the future work.

2 INEX 2014 Tweet Contextualization Track

The Initiative for the Evaluation of XML Retrieval (INEX) is an established evaluation forum for XML information retrieval (IR) [17]. In 2014, tweet contextualization INEX task at CLEF, aims "*given a new tweet, the system must provide some context about the subject of the tweet, in order to help the reader to understand it. This context should take the form of* **a readable summary, not exceeding 500 words, composed of passages from a provided Wikipedia corpus.**"[4]

[4] `https://inex.mmci.uni-saarland.de/tracks/qa/`

Like in iNEX Question Answering track 2011, 2012 and 2013, the present task is about contextualizing tweets, i.e. answering questions of the form "What is this tweet about?" using a recent cleaned dump of the Wikipedia[5]. As organizers claim, the general process involves three steps:

- Tweet analysis.
- Passage and/or XML elements retrieval.
- Construction of the answer.

Then, a relevant passage segment contains:

- Relevant information but
- As few non-relevant information as possible (the result is specific to the question).

2.1 Document Collection

The corpus has been rebuilt in 2013 from a dump of the English Wikipedia from November 2012. All notes and bibliographic references were removed to facilitate the extraction of plain text answers. (Notes and bibliographic references are difficult to handle). Organizers kept only non empty Wikipedia pages (pages having at least one section).

2.2 Tweets set

For the Track 2014, a set of 240 tweets in English have been selected by the organizers from CLEF RepLab 2013 together with their related entity. The tweets have >= 80 characters and do not contain urls in order to focus on content analysis.

In the CLEF-INEXorganizers words: "RepLab provides several annotations for tweets, we selected three types of them: the category (4 distinct), an entity name from the wikipedia (61 distinct) and a manual topic label (235 distinct). The entity name should be used as an entry point into wikipedia or DbPedia and gives the contextual perspective. The usefulness of topic labels for this automatic task is an open question at this moment because of their variety".

3 Summarization System

3.1 Cortex Summarizer

Cortex [18, 19] is a single-document extract summarization system. It uses an optimal decision algorithm that combines several metrics. These metrics result from processing statistical and informational algorithms on the document vector space representation.

[5] See the official CLEF-INEX 2014 Tweet Contextualization Track Website: `https://inex.mmci.uni-saarland.de/tracks/qa/`.

The INEX 2014 Tweet Contextualization Track evaluation is a real-world complex question (called long query) answering, in which the answer is a summary constructed from a set of relevant documents. The documents are parsed to create a corpus composed of the query and the the multi-document retrieved by a Perl program supplied by INEX organizers[6]. This program is coupled to Indri system[7] to obtain for each query, 50 documents from the whole corpus.

The idea is to represent the text in an appropriate vectorial space and apply numeric treatments to it. In order to reduce complexity, a preprocessing is performed to the question and the document: words are filtered, lemmatized and stemmed. The Cortex system uses 11 metrics (see [20, 19] for a detailed description of these metrics) to evaluate the sentence's relevance.

By example, the topic-sentence overlap measure assigns a higher ranking for the sentences containing question words and makes selected sentences more relevant. The overlap is defined as the normalized cardinality of the intersection between the query word set T and the sentence word set S.

$$\text{Overlap}(T, S) = \frac{card(S \cap T)}{card(T)} \tag{1}$$

The system scores each sentence with a decision algorithm that relies on the normalized metrics. Before combining the votes of the metrics, these have been split into two sets: one set contains every metric $\lambda^i > 0.5$, while the other set contains every metric $\lambda^i < 0.5$ (values equal to 0.5 are ignored). We then compute two values α and β, which give the sum of distances (positive for α and negative for β) to the threshold 0.5 (the number of metrics is Γ, which is 11 in our experiment):

$$\alpha = \sum_{i=1}^{\Gamma} (\lambda^i - 0.5); \quad \lambda^i > 0.5 \tag{2}$$

$$\beta = \sum_{i=1}^{\Gamma} (0.5 - \lambda^i); \quad \lambda^i < 0.5 \tag{3}$$

The value given to each sentence s given a query q is calculated with:

$$\begin{aligned} &\text{if}(\alpha > \beta) \\ &\qquad \text{then} \;\; \text{Score}(s, q) = 0.5 + \tfrac{\alpha}{\Gamma} \\ &\qquad \text{else} \;\; \text{Score}(s, q) = 0.5 - \tfrac{\beta}{\Gamma} \end{aligned} \tag{4}$$

The Cortex system is applied to each document of a topic and the summary is generated by concatenating higher score sentences.

[6] See: `http://qa.termwatch.es/data/getINEX2011corpus.pl.gz`

[7] Indri is a search engine from the Lemur project, a cooperative work between the University of Massachusetts and Carnegie Mellon University in order to build language modelling information retrieval tools. See: `http://www.lemurproject.org/indri/`

4 Artex

ARTEX[8] computes the score of each sentence by calculating the inner product between a sentence vector, an *average pseudo-sentence* vector (the "global topic") and an *average pseudo-word* vector (the "lexical weight"). The summary is generated concatenating the sentences with the highest scores.

An average document vector represents the "global topic" of all sentences vectors is constructed. The "lexical weight" for each sentence, i.e. the number of words in the sentence, is obtained. A score for each sentence is calculated using their proximity with the "global topic" and their "lexical weight". Let $\boldsymbol{s}_\mu = (s_{\mu,1}, s_{\mu,2}, \ldots, s_{\mu,N})$ be a vector of the sentence $\mu = 1, 2, \ldots, \rho$. The *average pseudo-word* vector $\boldsymbol{a} = [a_\mu]$, was defined as the average number of occurrences of N words used in the sentence $\boldsymbol{s}_\mu$:

$$a_\mu = \frac{1}{N} \sum_j s_{\mu,j} \tag{5}$$

and the *average pseudo-sentence* vector $\boldsymbol{b} = [b_j]$ as the average number of occurrences of each word j used through the ρ sentences:

$$b_j = \frac{1}{\rho} \sum_\mu s_{\mu,j} \tag{6}$$

The weight of each sentence is calculated as follows:

$$\omega(\boldsymbol{s}) = (\boldsymbol{s} \times \boldsymbol{b}) \ \times \boldsymbol{a} \tag{7}$$

4.1 Reg Summarizer

We create a graph $G = (V, A)$ where S vertices represent sentences and A the set of edges. An edge between two vertices is created if the corresponding sentences have at least one word in common. An adjacency matrix is constructed from the matrix $S_{[\mathrm{P=sentences} \times \mathrm{N=words}]}$ as follows: If the element $S_{i,k} = 1$ of S matrix (in the phrase i the word k is present), we check the k column. If the element $S_{j,k} = 1$ we put 1 in $a_{i,j}$ of the adjacency matrix A, which means that i and j sentences share the word k. To extract the heaviest sentence, a variant of tree problem maximum weight has been proposed. The weights are on the vertices; not on the edges. We have built an algorithm inspired on the Kruskal's algorithm [21]. The proposed algorithm works as follows:

- generate the adjacency matrix $A_{[P \times P]}$;
- calculating the weight of the vertices, i.e. the sum of the incoming edges of the vertex;
- calculate the degree of each vertex: i.e. the number of shared words with the other sentences.

[8] In French, ARTEX is *Autre Résumeur de TEXtes.*

The adjacency matrix $A_{[P \times P]}$ is generated from the VSM model:

$$a_{ij} = \begin{cases} 1 & \text{if a word used by the sentence } i \text{ is also used by the sentence } j \\ 0 & \text{elsewhere} \end{cases}$$

The solution is based on a calculation greedy search paths. The algorithm REG performs the following steps:

1. Select the vertex heavy v_0, and put it in T. It will be called **root**. The root is chosen whose degree is $>= 2$.
2. Add to T the heavy neighbor of v_0. It will choose among those who are not part of T.
3. Repeat 2 until k have the required vertices.
4. Return the path T.

5 Experiments Settings and Results

In this study, we used the document sets made available during the Initiative for the Evaluation of XML retrieval (INEX)[9], in particular on the INEX 2012 Tweet Contextualization Track.

The strategy of our three summarizer systems to deal multi-document summary problem is quite simple: first, a long single document D is formed by concatenation of all $i = 1, ..., n$ relevant documents provided by Indri engine: $d_1, d_2, ...d_n$. The first line of this multi-document D is the tweet T. Each summarizer extracts of D the most relevant sentences following T. Then, this subset of sentences is sorted by the date of documents d_i. The summarizer adds sentences into the summary until the word limit is reached. To evaluate the performance of eaxh system on INEX tweet contextualization track, we used the online package available from CLEF-INEX website[10].

6 Results

Table 1 shows the official results of Informativeness based on sentences. The performances (rank) of our summarizers are: Cortex (Run 356)=9/14, Artex (Run 357)=10/14 and Reg (Run 358)=12/14.

Table 2 shows the official results of Informativeness based on Noun Phrases (NP). The performances (rank) of our summarizers are: Cortex (Run 356)=9/14, Artex (Run 357)=10/14 and Reg (Run 358)=12/14.

Table 3 shows the official results of manual evaluation of CLEF-INEX 2014 contextualization task. The performances (rank) of our summarizers are: Cortex (Run 356)=2/14, Artex (Run 357)=3/14 and Reg (Run 358)=1/22. The results must be interpreted as follow:

[9] `https://inex.mmci.uni-saarland.de/`
[10] `http://qa.termwatch.es/data/`

Table 1. Informativeness based on sentences

Rank	Participant	Unigrams	Bigrams	skipgrams
1	ref2013	0.705	0.794	0.796
2	ref2014	0.7528	0.8499	0.8516
...		...		...
9	356 (Cortex)	0.8415	0.9696	0.9702
10	357 (Artex)	0.8539	0.97	0.9712
...		...		...
12	358 (Reg)	0.8731	0.9832	0.9841
...		...		...

Table 2. Informativeness based on NPs

Rank	Participant	Unigrams	Bigrams	skipgrams
1	ref2013	0.7468	0.8936	0.9237
2	ref2014	0.7784	0.917	0.9393
...		...		...
9	356 (Cortex)	0.8477	0.971	0.9751
10	357 (Artex)	0.8593	0.9709	0.9752
...		...		...
12	358 (Reg)	0.8816	0.984	0.9864
...		...		...

- Readable: % of passages considered as readable (Non trash)
- Syntax % of passages without syntax or grammatical errors
- Diversity % of non redundant passages
- Structure % of non breaking anaphora passages

7 Conclusions

In this paper we have presented three statistical summarizer systems applied on CLEF-INEX 2014 Tweet Contextualization Track. The first one, Cortex is based on the fusion process of several different sentence selection metrics. The decision algorithm obtains good scores on the INEX 2014 Tweet Contextualization Track (the decision process is a good strategy without training corpus). The second one, Artex is based on the inner product of main topic and pseudo-words vectors. The third system is Reg, a graph-based summarizer. Our three summarizers have obtained very good results in manual evaluations. Reg is the better system in terms of readability, syntax, diversity and structure manual evaluations. We show that a simple statistical summarizers without knowledge obtains good performances in this complex summarization and tweet contextualization task.

Table 3. Readability results for our systems (runs 356-358)

Rank	System (Run)	Readability	Syntax	Diversity	Structure	Average
1	**Reg (358)**	**94.82%**	**87.31%**	**72.17%**	**93.10%**	**86.85%**
2	Cortex (356)	95.24%	85.19%	70.31%	92.40%	85.78%
3	Artex (357)	94.88%	82.53%	71.34%	91.58%	85.08%
...			...			...
6	Ref'13	91.74%	69.82%	60.52%	85.80%	76.97%
7	Ref'12	91.39%	69.58%	60.67%	85.56%	76.80%
...			...			...
14	(370)	90.10%	68.30%	53.83%	80.70%	73.23%

References

1. ANSI. American National Standard for Writing Abstracts. Technical report, American National Standards Institute, Inc., New York, NY, 1979. (ANSI Z39.14.1979).
2. J.M. Torres-Moreno. *Resume automatique de documents : une approche statistique*. Hermes-Lavoisier, 2011.
3. H. P. Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159, 1958.
4. H. P. Edmundson. New Methods in Automatic Extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.
5. I. Mani and M. Mayburi. *Advances in automatic text summarization*. The MIT Press, U.S.A., 1999.
6. Gregory Salton. *The SMART Retrieval System - Experiments un Automatic Document Processing*. Englewood Cliffs, 1971.
7. Gregory Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
8. I. Da Cunha, S. Fernandez, P. Velázquez Morales, J. Vivaldi, E. SanJuan, and J.M. Torres-Moreno. A new hybrid summarizer based on vector space model, statistical physics and linguistics. In *MICAI 2007: Advances in Artificial Intelligence*, pages 872–882. Springer Berlin/Heidelberg, 2007.
9. J.M. Torres-Moreno, P. Velazquez-Morales, and JG. Meunier. Condensés automatiques de textes. *Lexicometrica. L'analyse de données textuelles : De l'enquête aux corpus littéraires*, Special(www.cavi.univ-paris3.fr/lexicometrica), 2004.
10. C. Jacquemin and P. Zweigenbaum. Traitement automatique des langues pour l'accès au contenu des documents. *Le document en sciences du traitement de l'information*, 4:71–109, 2000.
11. Jose Abracos and Gabriel Pereira Lopes. Statistical Methods for Retrieving Most Significant Paragraphs in Newspaper Articles. In Inderjeet Mani and Mark T. Maybury, editors, *ACL/EACL97-WS*, Madrid, Spain, July 11 1997.
12. Simone Teufel and Marc Moens. Sentence Extraction as a Classification Task. In Inderjeet Mani and Mark T. Maybury, editors, *ACL/EACL97-WS*, Madrid, Spain, 1997.
13. Eduard Hovy and Chin Yew Lin. Automated Text Summarization in SUMMARIST. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 81–94. The MIT Press, 1999.

14. Julian Kupiec, Jan O. Pedersen, and Francine Chen. A Trainable Document Summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, 1995.
15. Anastasios Tombros, Mark Sanderson, and Phil Gray. Advantages of Query Biased Summaries in Information Retrieval. In Eduard Hovy and Dragomir R. Radev, editors, *AAAI98-S*, pages 34–43, Stanford, California, USA, March 23–25 1998. The AAAI Press.
16. Judith D. Schlesinger, Deborah J. Backer, and Robert L. Donway. Using Document Features and Statistical Modeling to Improve Query-Based Summarization. In *DUC'01*, New Orleans, LA, 2001.
17. Shlomo Geva, Jaap Kamps, Ralf Schenkel, and Andrew Trotman, editors. *Comparative Evaluation of Focused Retrieval - 9th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2010, Vugh, The Netherlands, December 13-15, 2010, Revised Selected Papers*, volume 6932 of *Lecture Notes in Computer Science*. Springer, 2011.
18. J.M. Torres-Moreno, P. Velázquez-Morales, and J. Meunier. *CORTEX, un algorithme pour la condensation automatique de textes.* In *ARCo*, volume 2, page 365, 2005.
19. J.M. Torres-Moreno, P.L. St-Onge, M. Gagnon, M. El-Bèze, and P. Bellot. Automatic summarization system coupled with a question-answering system (qaas). *in CoRR*, abs/0905.2990, 2009.
20. J.M. Torres-Moreno, P. Velazquez-Morales, and J.G. Meunier. *Condensés de textes par des méthodes numériques. JADT*, 2:723–734, 2002.
21. R. Gould. *Graph Theory.* The Benjamin/Cummings Publishing Company,Inc, 1988.