University of Alicante at the NTCIR-12: Mobile Click

Fernando Llopis
University of Alicante, Spain
fernando.llopis@ua.es

Elena Lloret
University of Alicante, Spain
elloret@dlsi.ua.es

Jose M. Gomez
University of Alicante, Spain
jmgomez@ua.es

ABSTRACT

This paper describes the first participation of processing natural language group of the University of Alicante in Mobile Click Task of NTCIR 12. Our approach is based on the combination of tools developed in our research group: IR-n, a passage retrieval system; COMPENDIUM, a summarization generator; and a new approach based on Principal Component Analysis, another type of summarizer.

In our first participation we focused on the iUnit Ranking Subtask, although we have made an attempt on the iUnit Summarization Subtask.

Team Name

ALICA

Subtasks

iUnit Ranking Subtask (English) iUnit Summarization Subtask (English)

Keywords

Information Retrieval, Passage Retrieval, Summarization, MobileClick, NTCIR, COMPENDIUM, PCA, IR-n

1. INTRODUCTION

The current technology society requires huge amount of relevant information. Specialists, professionals, and all kind of people try to achieve the information that they need. However, information about any topic is increasing every day and to select the relevant one is an arduous task. It demands to waste more and more time trying to collect and to summarize all information we receive.

In this social context, systems which are able to filter, reduce and sort the relevant information from a given question are very interesting and its research is a necessity.The purpose of this task is to select , order and if necessary extract the most relevant information in order to focus the user in a useful but size limited information.

In our first participation in the Mobile Clik task, we used a passage retrieval system (IR- N) which was developed by [5] and two summarizers such as the COMPENDIUM [6] and Other based on PCA, Principal Component Analysis [10].

The core of our approach was developed using COMPENDIUM or semantic PCA-based summarization systems, although we had to use the IR-n system to meet the objectives of the task, mainly for ordering the sentences according to their relevance since the summarizers provide the selected sentences in the same order as the input files.

The main objective of this article is to explain the main features of IR-n systems and the summarizers systems and how they have been combined.

The remaining of the paper is organized as follows. In Section 2, we explain our COMPENDIUM summarization system. Section 3 describes the PCA technique. In Section 4 we review the IR-n system. Furthermore, in Sections 5 and 6 we describe the different subtasks which we have participated, and the results obtained (Section 7). Finally, the main conclusions are outlined in Section 8.

2. COMPENDIUM SUMMARIZER

COMPENDIUM is an automatic text summarisation tool that produces generic informative extracts from single or multiple documents. For the identification, selection and extraction of the most relevant information, different techniques are employed trough a pipeline of five stages. Figure 1 depics a graphical overview of the stages):

- **Surface linguistic analysis**. This stage pre-process the text carrying out a basic linguistic analysis, using external state-of-the-art tools and resources. This pre-processing includes sentence segmentation, tokenisation, part-of-speech tagging and syntactic analysis, stemming, and stopword identification and removal.

- **Redundancy detection**. The aim of this stage is to identify redundant information in the source documents, in order not to include it in the summary. For this purpose, Textual Entailment (TE) has been shown to be appropriate for this stage, since it determines whether the meaning of a text snippet can be inferred from another one [4].

  In order to illustrate this objective, we provide the following examples, taken from the RTE corpora[1]. As it can be seen the first example shows a true entailment relation, whereas the second example shows a false entailment.

[1]http://pascallin.ecs.soton.ac.uk/Challenges/RTE3/

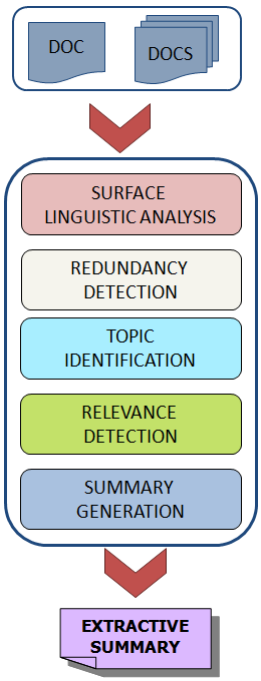

Figure 1: Overview of the COMPENDIUM approach.

Pair id=50 (entailment = true)
T: Edison decided to call "his" invention the Kinetoscope, combining the Greek root words "kineto" (movement), and "scopos" ("to view").
H: Edison invented the Kinetoscope.
Pair id=18 (entailment = false)
T: Gastrointestinal bleeding can happen as an adverse effect of non-steroidal anti-inflammatory drugs such as aspirin or ibuprofen.
H: Aspirin prevents gastrointestinal bleeding.

In our text summarization approach, the entailment relationship between two sentences is computed through an iterative process, discarding the last one, if they both contain a true entailment. This means that the meaning of such sentence is already embedded in the previous one, and therefore we would avoid the inclusion of repeated information. It is worth stressing upon the fact that the order in which the entailment relationships are computed is the same order the sentences have in the original documents. In this manner, we ensure that the coherence of the resulting summary is not highly affected after this stage. As a result of this stage, a set of sentences from the text which do not hold an entailment relation with any other is obtained, thus passing this set of sentences through the next stages of the process.

It is worth noting that this stage is optional. It can be activated or de-activated depending on the degree of redundancy that could appear in the texts. For instance, when generating summaries from a single document, this stage is not necessary, since the document may not contained repeated information. By de-activating it, we make the summarization process be faster.

- **Topic identification**. The objective of this stage is to determine the main topics of the document/s to be summarised.

  In COMPENDIUM, the topics of a document are represented by the frequency of the terms it contains. Following this statement, we assume that the most frequent terms of a document are indicative of the topics included in it. Therefore, sentences containing such topics (i.e., frequent terms) will be scored higher, as it will be further explained in the *Relevance Detection* stage.

- **Relevance detection**. At this stage, a weight is computed and assigned to each sentence, depending on how relevant it is within the text.

  This weight takes the term frequency computed in the previous stage, and it combines it with another feature based on *The Code Quantity Principle* (CQP) [3], assuming that: (1) a larger chunk of information is given a larger chunk of code; (2) the less predictable information, the more coding material; and (3) the more important information, the more coding material. The idea behind this theory is that when an item provides a specific information, it has to be assigned with a coding that would be more or less stressed according to the degree of relevance that such information has within the text. In other words, the most important information within a text will contain more lexical elements, and therefore it will be expressed by a high number of units (for instance, syllables, words or phrases) [1]. Taking this theory into account, a coding element can range from characters to phrases. COMPENDIUM bases its analysis on noun-phrases, because a noun-phrase is the syntactic structure which allows more flexibility in the number of elements it can contain (pronouns, adjectives, or even relative clauses). Moreover, it is able to carry more or less information (words) according to what the writer wants to express [1].
  For instance, if we take these two sentences as example:
  S_1: The Spanish Academy of Motion Pictures Arts and Sciences presented an honorific award for the best actor.
  S_2: The Academy presented an honorific award.
  In this case, S_1 contains more information than S_2. Although at a first sight, the second sentence might be more appropriate for TS, since it reflects the same facts of the first one but in a shorter manner, the first one contains more details, and this would lead to more informative summaries, which is the purpose of our TS process.

  For computing the relevance of a sentence, both the topics identified in the previous stage and the CQP are taken into account, and determines the relevance of each sentence by means of Formula 1.

$$r_{s_i} = \frac{1}{\#NP_i} \sum_{w \in NP} |tf_w| \quad (1)$$

where:
r_{s_i} = is the relevance of sentence i,

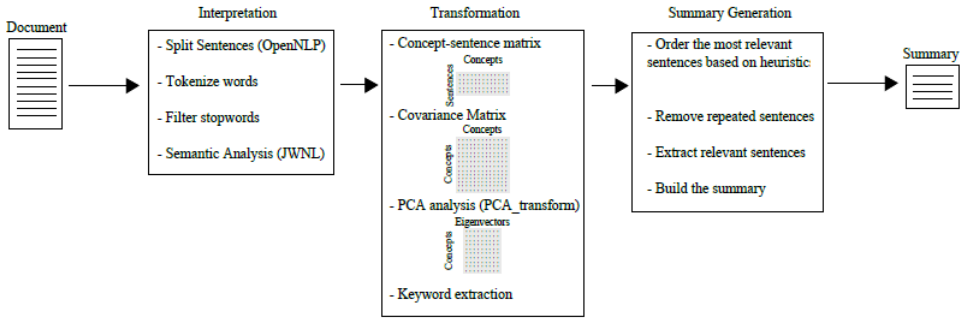

Figure 2: Our semantic PCA-based approach for generic extractive summarization

$\#NPi$ = number of noun-phrases contained in sentence i,
tf_w = frequency of word w that belongs to the sentence's noun-phrase.

In order to identify noun-phrases within a sentence, the *BaseNP Chunker*[2], is employed, which achieves recall and precision rates of roughly 93% for base noun-phrase chunks, and 88% for more complex chunks [8], thus being suitable for our purposes.

- **Summary generation**. The objective of this stage is to generate a summary of a specific length. This length is expressed in the form of a compression rate (i.e., the percentage of information the summary contains with respect to the source document). Given a compression rate, the most relevant sentences (i.e.,, the ones with higher scores) will be extracted to form the final summary up to such desired length. In order to minimize potential problems with the coherence of the produced summary, the selected sentences will be finally presented in the same order as they are in the source document.

3. SEMANTIC PCA-BASED SUMMARIZER

This summarization approach follows a generic flow consisting of three stages: i) *interpretation*; ii) *transformation*; and iii) *summary generation*. Figure 2 provides an overview of the approach, which is next explained in more detail.

- **Interpretation: Concept Identification**. First, a linguistic text preprocessing is necessary to proceed with the creation of the concept-sentence matrix. Once the input is splitted into sentences, using the OpenNLP Java library[3], each of them is tokenized to subsequently filter stopwords. Afterwards, a semantic analysis is applied to each word in order to identify concepts. For this, Wordnet [7] was employed to perform the semantic analysis using its JWNL Java library[4]. WordNet is a lexico-semantic English resource that groups words into sets of synonyms called synsets, providing information about the semantic relationships between them. In our approach, it is used to infer existing sets of synonyms in the documents, thus working with concepts instead of terms.

  For identifying concepts, the process searches for the first synset of each word in the document. The first synset Wordnet returns correspond to the most frequent sense of that word, and therefore it is the most probable meaning. If two words have the same first synset, we will assume that they are synonyms and their occurrences will be added together. For example, *detonation* and *explosion* are different words but their Wordnet's first synsets are the same (*07323181*), so we keep them as a single concept in the concept-sentence matrix.

  At the end of this stage, the text is prepared to compute the concept-sentence matrix and apply PCA technique, as it is next explained.

- **Transformation: PCA for Key Information Detection**. PCA is a statistical technique focused on the synthesis of information to compress and interpret the data [2]. For large volumes of data, the aim of this algorithm is to find a set of patterns or trends to reduce the dimensionality in the input data set. PCA tries to create projections of the input samples in a subspace of a smaller dimension by finding linear combinations of the original data. Those linear combinations are constructed in order of importance in terms of the total variability of the sample collected. The covariance matrix is computed to obtain the principal components (eigenvectors) and its corresponding weight (eigenvalue). PCA returns a matrix in which the eigenvectors are the columns and the rows are the variables of the covariance matrix. The eigenvectors are composed by the contribution of each variable, which determines the importance of the variable in the eigenvector. Moreover, the eigenvectors are derived in decreasing order of importance determined by the eigenvalue. In this manner, an eigenvector with high eigenvalue carries a great amount of information. Therefore, the first eigenvectors collect the major part of the information extracted from the covariance matrix.

  In our approach, PCA is applied using the PCA_trans-

[2]This resource is free available in ftp://ftp.cis.upenn.edu/pub/chunker/
[3]https://opennlp.apache.org/
[4]http://sourceforge.net/projects/jwordnet

form Java library[5] to process the covariance matrix from the concept-sentence matrix. In the concept-sentence matrix, the concepts (nouns, verbs, adverbs, and adjectives) are considered as variables (columns), whereas the sentences are the observations of the matrix (rows).

After applying PCA to the covariance matrix, for each eigenvector, the concept(s) with the highest value is/are extracted. These concepts are ordered by decreasing order of relevance, and will be used for selecting key sentences.

- **Summary Generation**. From the previous stages, the matrix with the eigenvectors from PCA is obtained; however, an important stage in any extractive summarization process is to finally determine and select the specific sentences that will constitute the summary to be used by users or other processes. Taking into account the concept with the highest value for each eigenvector from the PCA matrix, select and extract: *all the sentences in which at least two concepts appear. We therefore prioritize the importance of the concepts inside the sentence, rather than the highest number of concepts that a sentence may contain.*

  It is worth mentioning that if we found different concepts with the same highest value for the same eigenvector, we would extract the corresponding sentences for all these concepts. In the same manner, if a concept was represented by several synonyms, we would extract the corresponding sentences for each of these synonyms.

  Such strategy provide us with the relevance of the sentences in decreasing order, that will be chosen for building the summary until the desired length is reached. Moreover, in this approach, redundancy is avoided by not allowing the inclusion of repeated sentences, if these have already been selected.

4. IR-N SYSTEM

Passage Retrieval is an alternative to traditional document-oriented Information Retrieval. These systems use contiguous text fragments (or passages), instead of full documents, as basic unit of information. IR-n system [5] is a passage retrieval system that use groups of contiguous sentences as unit of information.

The system proposed has the main following features:

1. A document is divided into passages that are made up by a number N of sentences.

2. Passages overlap. In order to avoid splitting sentences with relevant information in different but continuous passages, we formed the passages using the overlap technique. That means, the first passage contains from sentence *1* to *N*, second passage contains from sentence *2* to $N+1$, and so on.

3. The similarity between a passage p and a query q is computed as follows:

$$Passage\_similarity = \sum_{t \in p \wedge q} W_{p,t} * W_{q,t} \qquad (2)$$

Where

$W_{p,t} = log_e(f_{p,t} + 1)$,

$f_{p,t}$ is the number of appearances of term t in passage p,

$W_{q,t} = log_e(f_{q,t} + 1) * idf$,

$f_{q,t}$ represents the number of appearances of term t in question q,

$idf = log_e(n/f_t + 1)$,

n is the number of documents of the collection and

f_t is the number of documents term t appears in.

As it can be observed, this formulation is similar to the cosine measure defined in [9]. The main difference is that length normalisation is omitted.

In the context of the NTCIR tasks that we participated we used IR-n to sort the sentences returned by summarizer systems in order to locate the most relevant sentences and place them in the top positions of the resulting rank. This step was necessary since the summarization systems directly generated a fragment of text (the summary) composed by significant sentences of the original documents, but in the same order as their appeared to ensure the coherence of the generated text. However, in this task, focused on information retrieval, was more important the ranking rather than the coherence.

5. IUNIT RANKING SUBTASK

"The iUnit ranking subtask is a task where systems are expected to rank a set of pieces of information (iUnits) based on their importance for a given query. This subtask was devised to enable *componentized* evaluation, where we can separately evaluate the performance of estimating important information pieces and summarizing them into two-layers."

We have a set of queries and a set of iUnits (each one represented by a sentence). The aim is to obtain a list with the iUnits ordered by the relevance with respect to the query.

5.1 Training process

We developed several experiments using the test collection in order to optimize the system's performance.

As a baseline system, we selected the passage retrieval IR-n system. We developed experiments with COMPENDIUM and semantic PCA-based systems. COMPENDIUM was configured to detect or not redundant information using textual entailment module. Moreover, we experimented with three compression ratios for summarization: 10%, 20% and 40%.

5.1.1 Baseline: IR-N System

Our baseline has been used to perform a single task information retrieval. Each query entry system was the concatenation of all iUnits.

For instance, if query 1 was "*1C2-E-0001 michael jackson death*" and the iUnits were:

[5]https://github.com/mkobos/pca_transform

- 1C2-E-0001 1C2-E-0001-0001 family concerned about murray role
- 1C2-E-0001 1C2-E-0001-0002 giving singer nightly doses of propofol
- 1C2-E-0001 1C2-E-0001-0003 murray first met jackson in las vegas

We would build the following question:
1C2-E-0001 family concerned about murray role. giving singer nightly doses of propofol. murray first met jackson in las vegas.
Then, for each iUnit we obtained the relevance Rel_{IRn} score as we can see in Figure 3.

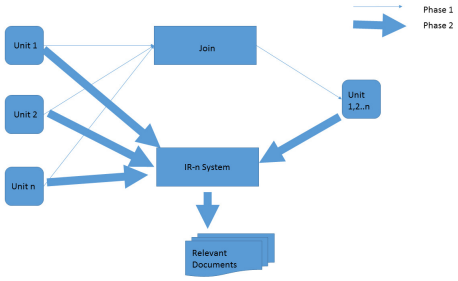

Figure 3: Baseline: IRn System.

5.1.2 *Runs with Text Summarization systems*

We generated a document with the concatenation of all iUnits. After that, we apply the summarization system (COMPENDIUM or semantic PCA-based) with 3 different compression ratios (10%, 20% and 40%) as we can see in Figure 4.

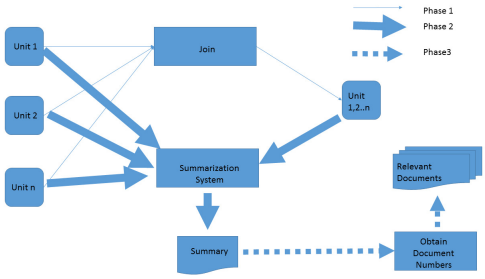

Figure 4: Runs with Summarization systems

Each iUnit selected by the Summarization system has a relevance factor Rel_{Com} equal to 1. If a iUnit is not selected, it has a value of 0.
We also tested the redundancy detection module based on textual entailment of COMPENDIUM. The results of all experiments can be see in Table 1.
Once the results were analyzed a set of conclusions we obtained. They are summed up in the following points:

- The Textual Entailment did not improve the results.
- COMPENDIUM obtained better results than the semantic PCA-based system.

		Q-measure		
		10%	20%	40%
Baseline (IR-n)	**0.8621**			
COMPENDIUM		0.8196	0.8291	**0.8403**
Semantic PCA-based		0.8101	0.8040	0.8064
COMPENDIUM+TE		0.82010	0.8218	0.8246

Table 1: Results with summarization systems alone

- In these experiments, we obtained better results with the baseline system (IR-n) because the summarization systems do not order the iUnits by their relevance as required by the tasks.

5.1.3 *Combining IR-n + COMPENDIUM*

Summaries generated by COMPENDIUM do not sort the iUnits with respect to their query relevance. Therefore an additional run was submitted. This run aims to place at the top the iUnits selected by COMPENDIUM system but arranged according to the relevance that IR-n gave them.
For each iUnit, the following relevance is obtained:
$Rel_{def} = Rel_{IRn} + (Rel_{Com} * 1000)$
The results are shown in Table 2. As it can be seen, the combination of both systems yields better results than using only the systems in an independent way.

		Q-measure
Baseline (IR-n)	**0.8621**	
COMPENDIUM		**0.8403**%
COMPENDIUM + IRn		**0.8648**%

Table 2: Results for the combination IR-n+COMPENDIUM

6. IUNIT SUMMARIZATION SUBTASK

"The iUnit summarization subtask is defined as follows: Given a query, a set of iUnits, and a set of intents, generate a structured textual output. In MobileClick, more precisely, the output must consist of two layers. The first layer is a list of iUnits and links to the second layer, while the second layer consists of lists of iUnits. Each link must be one of the provided intents and be associated with one of the iUnit lists in the second layer."
For participating in this subtask, we performed three steps:

- **iUnits Selection**. The objective of this step is to determine the most relevant units to the query. We selected the four units most relevant using IR-n system.
- **Links Selection**. The objective of this step is to determine the most relevants links for the query. We selected the two links most relevant to the units using IR-n system. We searched for synonyms for the words in the link. And we used as a query the words in the link and the synonyms. The relevance valour for each link is the sum of the relevance for each unit.
- **Units in the Links Selection**. The objective of this step is to determine the most relevant units for each of the links selected. We chose the two units most relevant in each link.

	Q-measure
IRn + COMPENDIUM (*)	**0.9027**
TITEC	0.9003
UHYG	0.8994
ORG	0.8975
RISAR	0.8972
RISAR	0.8962
IRn	0.8959
COMPENDIUM	0.8934

Table 3: Results for Test iUnit Ranking Subtask

	Q-measure
TITEC	18.2596
ORG	16.8975
RISAR	16.047
ORG	14.1051
ORG	13.2689
IRn	8.4968

Table 4: Results for Test iUnit Summarization Subtask

7. RESULTS

7.1 iUnit Ranking Subtask

The results can be seen in Table 3. We obtained the best result with IRn+COMPENDIUM. However, this was not an official score because we could not finish the training on time.

7.2 iUnit Summarization Subtask

In Table 4, the results for this subtask are provided. We did not obtain a satisfactory results. This was due to the fact taht we did not have enough time to work on this task.

8. CONCLUSIONS

Given the current needs of current Internet users, mobile click task has enormous interest. Being capable of reducing the information that the user must process is an increasing demand. The Internet user wants to consult and get a short information but sufficiently proven.

Summarization systems are interesting, but require the integration of additional information to avoid duplication, as well as to sort out the information according to its relevance.

In this paper, we analyzed the improvement obtained when our passage retrieval system, called IR-n and a our summarization system called COMPENDIUM are combined. This improvement was evaluated on the Mobile Clik task within the NTCIR 12 forum.

Lack of time prevented us going deeper in the summarization task. Therefore, we understand that we have to work on improving the basic model used in the test.

As future work, we intend to work with the queries to obtain more information, so that we can improve our system's precision. Moreover, we need to investigate the effects of query expansion techniques over the intents. Finally, we are also trying to improve the ordering of sentences extracted by the summarization systems to improve the user experience at reading.

Acknowledgements

This research work has been partially funded by the University of Alicante, Generalitat Valenciana, Spanish Government and the European Commission through the projects, "Explotación y tratamiento de la información disponible en Internet para la anotación y generación de textos adaptados al usuario" (GRE13-15), "DIIM2.0: Desarrollo de técnicas Inteligentes e Interactivas de Minería y generación de información sobre la Web 2.0" (PROMETEOII/2014/001), TIN2015-65100-R, TIN2015-65136-C2-2-R, and SAM (FP7-611312).

9. REFERENCES

[1] A. Becker. Análisis de la Estructura pragmática de la cláusula en el español de Mérida (Venezuela). *Estudios de Lingüística del Español*, 17, 2002.

[2] E. Estellés Arolas, F. González Ladrón De Guevara, and A. Falcó Montesinos. Principal Component Analysis for Automatic Tag Suggestion. Technical report, 2010.

[3] T. Givón. *Syntax: A functional-typological introduction, II*. John Benjamins, 1990.

[4] O. Glickman. *Applied Textual Entailment*. PhD thesis, 2006.

[5] F. Llopis and J. L. Vicedo. IR-n system, a passage retrieval system at CLEF 2001. pages 244–252.

[6] E. LLORET and M. PALOMAR. COMPENDIUM: a text summarisation tool for generating summaries of multiple purposes, domains, and genres. *Natural Language Engineering*, 19(02):147–186, jul 2012.

[7] G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[8] L. A. Ramshaw and M. P. Marcus. Text Chunking Using Transformation-Based Learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, pages 82–94, 1995.

[9] G. A. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison Wesley, New York, 1989.

[10] M. Vicente, O. Alcón, and E. Lloret. The University of Alicante at MultiLing 2015: approach, results and further insights, nov 2015.