Short Message Contextualization

Liana Ermakova

Institut de Recherche en Informatique de Toulouse,
118 Route de Narbonne, 31062 Toulouse Cedex 9, France
ermakova@irit.fr

Abstract. The paper presents a novel approach to automatic multi-document summarization applied for short message contextualization. The proposed method is based on named entity recognition, part-of-speech weighting and sentence quality measuring. In contrast to previous research, we introduced an algorithm of smoothing from the local context. Our approach exploits topic-comment structure of a text. Moreover, we developed a graph-based algorithm for sentence reordering. The method was adapted to snippet retrieval and query expansion. The evaluation results on INEX and TREC data sets indicate good performance of the approach.

Keywords: Information retrieval, tweet contextualization, summarization, snippet, sentence extraction, readability, topic-comment structure

1 Introduction

The efficient communication tends to follow the principle of the least effort. According to this principle, **neither speakers nor hearers using a given language want to work any harder than necessary to reach understanding**. This fact led to the extreme compression of texts especially in electronic communication, e.g. microblogs, SMS, search queries. However, sometimes these texts are not self-content and need to be explained since understanding of them requires background knowledge, e.g. terminology, named entities or related facts. Therefore, the main motivation of this research is to help a user better understand a short message by providing a context. This context may be retrieved from an external source like Web or Wikipedia and in this case it may be considered as a multi-document query-biased summarization. Another application of the contextualization is retrieving of snippets allowing users to understand whether a document is relevant without reading it. Moreover, query expansion in a search engine may be also viewed as contextualization of the initial query.

This paper presents a novel general approach to contextualization task. The method was adapted to snippet retrieval and query expansion. The proposed approach is based on named entity recognition, part-of-speech weighting and sentence quality measuring. Usually, a sentence is viewed as a unit in summarization task. However, often a single sentence is not sufficient to catch it's meaning and even human beings need a context. Therefore, we introduce an algorithm to smooth a candidate sentence by it's local context. In contrast to [33],

we believe that a context does not provide redundant information, but allows to precise and extend sentence meaning. Neighboring sentences influence the sentence of interest, but this influence decreases as the remoteness of the context increases, which differs from the previous approaches where the dependence is considered to be binary [20]. Our algorithm takes advantage of topic-comment structure of sentences. The topic-comment structure have already got the attention of linguists in the 19-th century, however, it is hardly applied in information retrieval tasks. The retrieved sentences should be organized into a coherent text. If an extraction system deals with entire passages (which is our case), locally they may have higher readability than generated phrases since they are written by humans. Nevertheless, it is important to keep in mind the global readability of extracted passages. The only way to improve the readability of a text produced by an extraction system is to reorder the extracted passages. Thus, we introduce a novel graph-based algorithm for sentence reordering under the chronological constraints. Unlike Barzilay et al.s method [4], we do not search for the Hamiltonian path of maximal length, but for the minimal one. In our graph, the vertices correspond to the extracted passages and the edges represent the similarity measure between them.

The rest of the paper is organized as follows. Section 2 describes related works. Section 3 presents our method in general. Section 4 describes the INEX task. Section 5 contains the results and discusses them. Section 6 demonstrates other applications of the proposed approach. Section 7 concludes the paper.

2 Related works

The idea to contextualize short texts like micro-blogs or tweets is quite recent. Meij et al. mapped a tweet into a set of Wikipedia articles but in their work, no summary is provided to the user, rather a set of related links [18]. San Juan et al. went a step further and introduced tweet contextualization as an INEX task: participants had to contextualize tweets from the Wikipedia [24]. This implies text summarization. A summary is either an "extract", if it consists in the most important passages extracted from the original text, or an "abstract", if these sentences are re-written, generating a new text. Abstract generation is usually based on extraction which implies searching for relevant sentences [9].

Summarization can be either statistical, linguistic or mixed [16, 15]. Statistical methods can be referred to term frequency analysis [9, 23, 25] and machine learning [16]. Apparently, the first article on automated summarization was published by Luhn in 1958 [17] who proposed to order sentences by the number of the most frequent meaningful words. This approach was extended by taking into account inverse document/sentence frequency (IDF), sentence position in the text, key word and key phrase occurrence [9, 23, 25]. The multi-document summarization system SUMMARIST combined statistical approach with rules for word positions, key-phrases, e.g. *the most important, to conclude, to summarize* etc., and generalization of the related terms [16, 15].

Information useful for summarization may be found in semi-structured documents (e.g. in XML formats [1]), Wikipedia [19, 21], the context obtained through hyperlinks [8], the click-through data [27], or social networks (e.g. users comments and URLs posted on Twitter)[33].

Applying machine learning for summarization requires a corpus consisting of original texts and corresponding summaries. Text corpora provide much useful information on features which should be kept in a summary, how long a text should be, etc. [16, 23, 1].

Probabilistic graphical models are widely used for summarization. One of the most common models is conditional random fields [26, 33]. As other machine-learning methods, probabilistic graphical models need training data. Another very efficient model is Latent Dirichlet Allocation (LDA) [6]. LDA is a graphical topic model where a document is viewed as a mixture of topics and a topic is considered as a mixture of words [2]. Sentences are scored according to their probability to represent the topics. In the LexRank algorithm, a document is viewed as a graph where vertices correspond to the sentence and the edges represent the similarity measure between them [9]. Sentences are scored by expected probability of a random walker visiting each sentence [22]. In [22] edges correspond to the probability of two sentences to represent the same point of view. As LDA, weighted feature subset non-negative matrix factorization allows to obtain the most representative terms among the topics [31].

In the case of a subject related summary, like tweet contextualization, the subject may be considered as a query. Thus, a summary is made of the sentences relevant to the query [1]. A query can be expanded by synonyms from the WordNet, terms from headers or the most frequent words [1], Basic Elements (BEs) [12], or Wikipedia redirects [21]. The standard IR techniques of query expansion based on term co-occurrence are also applied in summarization task [13, 1].

Linguistic methods fall into several categories: (1) rule-based approaches, which may be combined with statistics [16, 15], (2) methods based on genre features, text structure etc. [5, 16, 25, 29] and (3) methods based on syntax analysis [5, 29]. One of the first summarizers was the domain-specific rule-based system SUMMONS which had an extraction component and a linguistic module for checking the syntax [23]. Rule-based approaches are very labor-intensive. Moreover, they are not flexible nor scalable. For those reasons their popularity decreased.

Different genres should be compressed at different rate, e.g. a news article may retain 25-30% of the original size while for a scientific paper this coefficient is about 3% [29]. Besides, one can take advantage of the text structure [25]. For example, in news the most important information is written at the beginning of an article, while scientific papers have an abstract [16]. Moreover, in scientific texts a sentence has a specific rhetorical status: research goals, methods, results, contribution, scientific argumentation or attitude toward other peoples work. A rhetorical status may be assigned according to matching to a linguistic pattern, position in the text, key words, grammatical features (verb tenses, modal verbs etc.) [28, 29]. As for news articles, multiple descriptions of the same events are

rather typical for them [5, 29]. So in the news articles the most important information tends to be mentioned several times [5]. However, this idea is very similar to the approached based on word frequencies.

[30] proposed to apply Manifold-Ranking algorithm for redundancy. The algorithm implies iterative selection of candidate sentences and is based on the assumption that sentences should provide different information and therefore they should not be similar. We argue the fact that sentences making a text should be as different as possible since reasonable sentence similarity is one of the major sign of text integrity and coherence. Therefore, in our approach we exclude only very similar sentences.

One of the clues to readability is sentence ordering [4]. In single-document summarization systems it is possible to use original sentence order. The idea was adopted by Majority Ordering algorithm for multi-document summarization. Subjects (sentences expressing the same meaning) T_i are organized into a directed graph were edges present the number of documents where T_i is followed by T_j and the best order corresponds to the Hamiltonian path of maximal length [4]. Another approach is to assign time stamp to each event and to order them chronologically. The use of chronological ordering is restricted to the news articles on the same topic [4]. Diversity topics in the news demand another way to arrange sentences extracted for multi-document summarization. Application of a text corpus provides the ground for improving readability. In this case the optimal order is found by the greedy algorithm maximizing the total probability [14]. In a narrative text verbs and adjectives play an important role in the semantic relations between sentences [3, 14].

3 Method Description

3.1 Sentence scoring

We assume that relevant sentences come from relevant documents. Document are retrieved by the Terrier platform[1] and parsed by Stanford CoreNLP[2] which integrates such tools as POS tagger and named entity recognizer. Then, we merged annotations obtained by parsers and Wikipedia tagging.

The sentence score $score(S)$ is estimated as the product of its quality $SntQual(S)$, smoothed relevance $R(S)$ and the score of the document $DocRel(d)$ from which it is extracted:

$$score(S) = DocRel(d) \times SntQual(S) \times R(S) \tag{1}$$

Sentence quality measure $SntQual(S)$ is used to avoid trash passages. We define it as the function of the lexical diversity $LexDiv(S)$, meaningful word ratio $Meaning(S)$ and punctuation score $PunctScore(S)$:

$$SntQual(S) = LexDiv(S) \times Meaning(S) \times PunctScore(S) \tag{2}$$

[1] terrier.org/

[2] nlp.stanford.edu/software/corenlp.shtml

Lexical diversity allows avoiding sentences that do not contain terms except those from a query. We define it as the number of different lemmas used within a sentence divided by the total number of tokens in this sentence. Meaningful word ratio over the total number of tokens in the sentence is aimed to penalized sentences that either have no sense at all or are not comprehensible without large context. $PunctScore(S)$ penalizes sentences containing many punctuation marks and is estimated by the formula:

$$PunctScore(S) = 1 - \frac{PunctuationMarkCount(S)}{TokenCount(S)} \tag{3}$$

where $PunctuationMarkCount(S)$ is a total number of punctuation marks in the sentence, and $TokenCount(S)$ is a total number of tokens in S. Thus, $PunctScore(S)$ shows the ratio of tokens which are not punctuation marks. Thus, we believe that a good sentence should have high ratio of different meaningful words and reasonable ratio of punctuation.

We introduced an algorithm for smoothing from the local context. We assumed that the neighboring sentences influence the sentence of interest, but this influence decreases as the remoteness of the context increases. In other words, the nearest sentences should produce more effect on the target sentence sense than others. We choose the simplest dependence model, namely the linear function. In this case $R(S)$ is calculated by the formulas:

$$R(S) = \sum_{i=-k}^{k} w_i \times r_i, \ \sum_{i=-k}^{k} w_i = 1 \tag{4}$$

$$w_i = \begin{cases} \frac{1-w_t}{k+1} \times \frac{k-|i|}{k} & 0 < |i| \leq k \\ w(S), & i = 0 \\ 0, & |i| > k \end{cases} \tag{5}$$

where $w(S)$ is the weight of the sentence S, w_i and r_i are respectively the weights and the prior scores of the sentences from the context of S of k length. If the sentence number in left or right context is less than k, their weights are added to the target sentence weight $w(S)$. This allows keeping the sum equal to one since otherwise a sentence with a small number of neighbors (e.g. the first or last sentences) would be penalized.

Prior scores of sentence r_i is a product of the cosine similarity measure sim_{uni} between the sentence and the query and the named entity similarity sim_{NE}:

$$r_i = sim_{uni} \times sim_{NE} \tag{6}$$

$$sim_{NE} = \frac{NE_{common} + NE_{weight}}{NE_{query} + 1} \tag{7}$$

where NE_{weight} is positive floating point parameter, NE_{common} is the number of NE appearing in both query and sentence, NE_{query} is the number of NE appearing in the query. NE_{weight} allows not to reject sentence without NE which can be still relevant. We add 1 to the denominator to avoid division by zero.

3.2 Topic-comment relationship in contextualization task

Linguistics establishes the difference between the clause-level topic and the discourse-level topic. The discourse-level topic refers to the notion of aboutness. While most information retrieval models make the assumption that relevant documents are about the query and that aboutness can be captured considering bags of words only, we rather consider a clause-level topic-comment structure. The topic (or theme) is the phrase in a clause that the rest of the clause is understood to be about, and the comment (also called rheme or focus) is what is being said about the topic. In simple English clause the topic usually coincides with the subject, however it is not a case of the passive voice. In most languages the common means to mark topic-comment relation are word order and intonation. Moreover, there exist special constructions to introduce the comment. However, the tendency is to use so-called topic fronting, i.e. to place topic at the beginning of a clause.

We hypothesize that topic-comment relationship identification is useful for contextualization. Quick query analysis provides evidence that an entity is considered as a topic, while tweet content refers rather to comment, i.e. what is said about the entity. Moreover, we assume that providing the context to an entity implies that this context should be about the entity, i.e. the entity is the topic, while the retrieved context presents the comment. We used these assumptions for candidate sentence scoring. We double the weight of sentences in which the topic contains the entity under consideration. Topic identification is performed under assumption of topic fronting. We simplify this hypothesis by assuming that topic should be place at the sentence beginning. Sentence beginning is viewed as the first half of the sentence.

We performed query preprocessing which differs over the runs:

- In order to link an entity and a tweet we combined the fields entity, topic and content into a single search query.
- The second way is to process fields entity and content as separate queries and then use the results obtained for the entity as a restriction to filter results retrieved for the tweet. Thus, the document retrieved by using the field content as a query are rejected if they do not coincide with top-ranked documents retrieved by using the field entity.

3.3 Sentence re-ordering

We propose an approach to increase global coherence of text on the basis of its graph model, where vertices represent sentences and edges correspond to the same cosine similarity measure as in searching for relevant sentences. If two relevant sentences are neighbors in the original text, they are considered as a single vertex. The hypothesis is that neighboring sentences should be somehow similar to each other and the total distance between them should be minimal. Firstly, we computed the similarity between sentences and reduced sentence ordering task to traveling salesman problem.

The traveling salesman problem (TSP) is an NP-hard problem in combinatorial optimization. Given a list of cities and their pairwise distances, the task is to find the shortest possible route that visits each city exactly once and returns to the origin city. In the symmetric case, TSP may be formulated as searching for the minimal Hamiltonian cycle in an undirected graph. Asymmetric TSP implies a directed graph. The obvious solution is to use brute force search, i.e. find the best solution among all possible permutations. The complexity of this approach is $O(n!)$ while other exact algorithms are exponential. Therefore, we chose the greedy nearest neighbor algorithm with minor changes. Since sentence ordering does not request to return to the start vertex and the start vertex is arbitrary, we tried every vertex as the start one and chose the best result, i.e. the start vertex giving the path of the minimal length.

However, this method does not consider chronological constraints. So, we modified the task and it gave us the sequential ordering problem (SOP). SOP "is a version of the asymmetric traveling salesman problem (ATSP) where precedence constraints on the vertices must also be observed" [11]. SOP is stated as follows. Given a directed graph, find a Hamiltonian path of the minimal length from the start vertex to the terminal vertex observing precedence constraints. Usually SOP is solved by the means of integer programming. Integer programming is NP-hard and these methods achieved only limited success. Therefore, we solved the problem as follows. Firstly, we ordered sentences with time stamps $s_1 - s_2 - ... - s_n$. Sentences without time stamp were added to the set $P = \{p_j\}_{j=\overline{1,m}}$. For each pair $s_i - s_{i+1}$ we searched for the shortest path passing through vertices from P. These vertices were removed from P and $i = i + 1$. If $i = n$, we searched for the shortest path passing through all vertices in P and the edge with the maximal weight was removed. The description of the algorithm is given in the figure 1.

4 Other applications

4.1 Snippet retrieval

A search engine returns to a user the immense volume of results that it is impossible to read. Therefore, to define whether a web page is relevant to a query without clicking a link, a search engine provides a user with snippets. A snippet is small text passages appearing under a search result extracted from the document content or its metadata. Ideally, a snippet provides the information a user is searching for. For the Snippet Retrieval Track we slightly modified the method applied for Tweet Contextualization by different sentence scoring and passage selection algorithms. We assumed that several features should be opposite to those applied in multi-document summarization, e.g. we do not penalize nominal sentences or we do not consider sentence ordering to be important for snippet retrieval. Moreover, we do not treat redundancy for snippets by the following reasons: (1) in the single-document summarization the probability of redundant information is much lower than in the multi-document one; (2) snippets are very short; (3) they should be generated very fast. We propose to use two algorithms

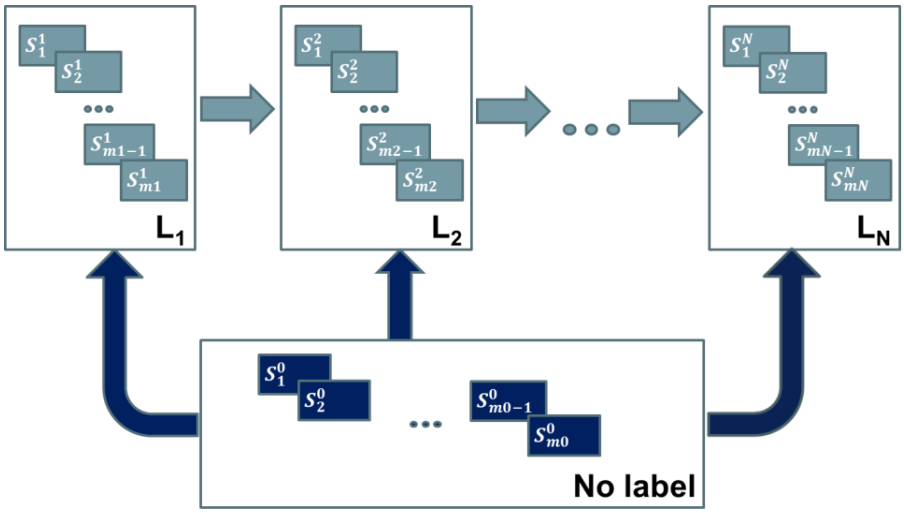

Fig. 1. Sentence reordering algorithm

for the candidate passage selection: dynamic programming approach to solve the knapsack problem and the moving window algorithm.

Knapsack Problem A snippet is limited up to 1-2 sentences (usually, 150-300 symbols). However, it should provide as much information about the underlying document as possible. Therefore, we can consider snippet generation task as selecting passages of the maximal total importance and the total weight no greater than a predefined threshold. This is a classical problem in combinatorial optimization, namely the knapsack problem. The knapsack problem is stated as follow: given a set of items, each with a weight and a value, find the subset of this set to pack the rucksack so that the total weight is less than or equal to a given capacity and the total value is as large as possible. As a weight, we consider the number of words in a sentence, and relevance represents a value. We are dealing with $0-1$ knapsack problem, which restricts the number of each kind of item to zero or one, since otherwise a snippet would have redundant information. We solve this problem by the basic dynamic programming algorithm $DP-1$ with an overall running time $O(nc)$ where n is the number of items and c is the knapsack capacity.

Moving Window There are two major drawbacks of the passage selection by applying the knapsack problem:

- If each sentence within a document were greater than a predefined threshold, the snippet would be an empty string.
- It has pseudo-polynomial time.

Therefore, we used a moving window algorithm (MW) to find the best score passage. At each step the first token is removed from a candidate passage and the tokens following the candidate passage are added while the new candidate total weight is no greater than a predefined threshold. The passage with the maximal score is selected as a snippet. Despite the most relevant information may occur in the too long sentences, snippets beginning in the middle of a sentence have lower readability. That is why, we penalize the snippets which do not starts at the beginning of a sentence. As opposed to the knapsack algorithm, MW is not suitable to tweet contextualization, as it is efficient only for very small extractive summaries. Summaries built by MW are exclusively made of consecutive sentences.

4.2 Query expansion

The key idea of the proposed method is to search the most appropriate terms for QE in the top ranked documents. Searching for the best-fit terms is based on ranking of terms and sentences. Both ranking procedures include local context analysis, i.e. analysis of neighboring sentences.

Our approach is underlain by the following hypotheses:

1. Terms for QE come from appropriate sentences (in general, this hypothesis is similar to those of RF). Right sentences should be of high quality and moreover they should match the query. The measure of sentence appropriateness is called sentence score and referred to $score(S)$ in the rest of the paper.
2. Good terms should have appropriate part of speech (POS) and high IDF. Not all POS are suitable for query expansion (e.g. functional words). Moreover, the most frequent terms are nouns. However, in some cases adjectives, verbs and numbers are indispensable. A good term should well distinguish documents from each other. POS weight and IDF may be considered as a query-independent term score.
3. The terms lying in the neighborhood of query terms are closer related to them than the remote ones.

We used a two-step local context analysis: for sentence scoring and for estimation of term importance. In previous works local context was viewed as a single document and it was opposed to the entire collection analysis (global context) [7, 32]. We consider local context in a stricter way, precisely we look not only to the whole document statistics, but also for terms surrounding the query terms.

All candidate terms are ranked according to the following metric:

$$w_{total}(t) = f(score(S), w_{pos}(t), IDF(t), importance(t, Q)) \tag{8}$$

where $score(S)$ is score of the sentence S containing t, $w_{pos}(t)$ is the weight of the POS of t, $IDF(t)$ is the inverse document frequency of the candidate term, $importance(t, Q)$ is a function of (1) the distance to the query Q terms, (2) their

weights, and (3) the likelihood of the candidate term to co-occur not by chance with the query terms in the top ranked documents.

$importance(t, Q)$ allows to find terms occurring in the neighborhood of important query terms.

Sentence quality is query-independent, while sentence weight $ws(S)$ shows how well a sentence matches a query.

In our method sentence weight depends on pseudo-relevance $DocRel(d)$ of the corresponding document d assigned by a search engine (i.e. document rank, score or their combination)

The next step of our method is to compute the importance of all terms in all sentences from RF:

$$importance(t, Q) = wd(t, Q) \times cooccurrence(t, Q) \tag{9}$$

$wd(t, Q)$ is a function of the distance from the candidate terms to the query Q and their weights, and $coocurence(t, Q)$ shows the likelihood of the candidate term to occur not by chance with the query terms in the top documents ranked according to the initial query.

5 Evaluation

We evaluated our approach on two INEX tracks: Tweet Contextualization 2011-2014 and Snippet Retrieval 2012-2013.

5.1 Tweet Contextualization Track

INEX Tweet Contextualization Track aims to evaluate systems providing context to a tweet. The context should be a readable summary up to 500 words extracted from a dump of the Wikipedia.

In 2011 our system showed the best results according the relevance judgment (see [10] for details). The system was based on TF-IDF cosine similarity measure, special weighting for POS, NE, structural elements of the documents, definitional sentences and the algorithm for smoothing from local context. In 2012 we modified our method by adding bigram similarity, anaphora resolution, hashtag processing, redundancy treatment and sentence reordering. However, we obtained lower results than in previous year. Therefore, in 2013 we decided to not consider bigram similarity, anaphora resolution, and redundancy treatment. We enriched our method by sentence quality measure based on Flesch reading ease test, lexical diversity, meaningful word ratio and punctuation ratio. We also used generalized POS (e.g. we merge regular adverbs, superlative and comparative into a single adverb group). In 2014 we introduced the topic-comment analysis. In this paper we will focus on the results demonstrated at INEX 2013-2014.

Summaries were evaluated according to their informativeness and readability.

Informativeness was estimated as the lexical overlap (*Unigrams, Bigrams* and *Skip bigrams* representing the proportion of shared unigrams, bigrams and

bigrams with gaps of two tokens respectively) of a summary with the pool of relevant passages extracted from the runs submitted by the participants. Official ranking was based on decreasing score of divergence with the gold standard estimated by skip bigrams:

$$Dis(S,T) = \sum_{t \in T} \frac{f_{T(t)}}{f_T} \times \left(1 - \frac{\min \log P, \log Q}{\max \log P, \log Q}\right) \quad (10)$$

where $P = \frac{f_{T(t)}}{f_T} + 1$ and $Q = \frac{f_{S(t)}}{f_S} + 1$, T is the set of terms in the pool of relevant passages, $f_{T(t)}$ is the frequency of a term t (*Unigrams, Bigrams* or *Skip bigrams*) in the pool, $f_{S(t)}$ is the frequency of a term t in a summary.

In 2013 the informativeness was estimated as the overlap of a summary with 3 pools of relevant passages:

- Prior set (PRIOR) of relevant pages selected by organizers (40 tweets, 380 passages).
- Pool selection (POOL) of the most relevant passages (1 760) from participant submissions for 45 selected tweets.
- All relevant texts (ALL) merged together with extra passages from a random pool of 10 tweets (70 tweets having 2 378 relevant passages).

We submitted 3 runs differing by sentence quality score and smoothing. Among automatic runs our best run 275 was ranked first (PRIOR and POOL) and second (ALL) over 24 runs submitted by all participants (see Table 1; our runs are set off in bold). It means that our best run is composed from the sentence of the most relevant documents. It is also obvious that ranking is sensitive to not only pool selection, but also choice of divergence. According to bigrams and skip bigrams our best run is 275, while according to unigrams the best run is 273. The runs 273 and 274 are quite close. In the run 273 each sentence is smoothed by its local context and first sentences from Wikipedia article which it is taken from. The run 274 has the same parameters except it does not have any smoothing. So, we can conclude that smoothing improves Informativeness. In our best run 275 punctuation score is not taken into account, it has slightly different formula for NE comparison and no penalization for numbers.

In 2014 participants should provide a context to 240 tweets in English from the perspective of the related entities. These tweets were collected by the organizers of CLEF RepLab 2013. They have at least 80 characters and do not contain URLs. A tweet has the following annotation types: the category (4 distinct), an entity name from the Wikipedia (64 distinct) and a manual topic label (235 distinct). The context has to explain the relationship between a tweet and an entity. As in previous years it should be a summary extracted from a Wikipedia dump. The organizers used 2 gold standards (1/5 of the topics):

- pool of relevant sentences per topic (SENT);
- pool of noun phrases (NOUN) extracted from these sentences together with the corresponding Wikipedia entry.

Table 1. Informativeness evaluation 2013

Run	All.skip	All.bi	All.uni	Pool.skip	Pool.bi	Pool.uni	Prior.skip	Prior.bi	Prior.uni
258	0,894	0,891	0,794	0,880	0,877	0,792	0,929	0,923	0,799
275	**0,897**	**0,892**	**0,806**	**0,879**	**0,875**	**0,794**	**0,917**	**0,911**	**0,790**
273	**0,897**	**0,892**	**0,800**	**0,880**	**0,875**	**0,792**	**0,924**	**0,916**	**0,786**
274	**0,897**	**0,892**	**0,801**	**0,881**	**0,875**	**0,793**	**0,923**	**0,915**	**0,787**

We submitted 3 runs:

- The first run (ETC) was performed by the system 2013. As a query three fields entity, topic and content were treated. An entity was treated as a single phrase.
- The second run (ETC_ENTITY) differed from ETC by double weight for sentences where the entity represented the topic.
- Unlike ETC, the third run (ETC_RESTR_NOENT) was based on document set restricted by entities (see the subsection 2.1 Preprocessing).

According to the evaluation performed on the pool of sentences, our runs ETC, ETC_ENTITY and ETC_RESTR_NOENT were classified 3-rd, 4-nd and 6-th; while according to the evaluation based on noun phrases, they got slightly better ranks, namely 2, 3 and 5 respectively. Thus, the best results among our runs were obtained by the system that merges fields entity, topic and content into a single query. The run #360 is better than our runs according to sentence evaluation; nevertheless, it showed worse results according to noun phrase evaluation. Our system is targeted on the nouns and especially named entities. This could provoke the differences in ranking with respect to sentences and noun phrases. The worst results were showed by the run based on entity restriction. This could be explained by the fact that filtering out the documents that are considered irrelevant to the entity may cause the big loss of relevant documents if they are not top-ranked according to entities. ETC_RESTR_NOENT demonstrated the worst results among our runs even in the case of noun phrases. We believe that this is caused by loss in recall since the importance of noun phrases is not evaluated, but filtering out some documents could have negative effect on noun phrase recall. The results of ETC and ETC_ENTITY are very close. However, topic-subject identification slightly decreased the performance of the system. Yet we believe that finer topic-comment identification procedure may ameliorate the results.

Table 2. Informativeness evaluation 2014

Run	SENT.uni	SENT.bi	SENT.skip	NOUN.uni	NOUN.bi	NOUN.skip
361	0.7632	0.8689	0.8702	0.7903	0.9273	0.9461
360	0.782	0.8925	0.8934	0.8104	0.9406	0.9553
ETC	**0.8112**	**0.9066**	**0.9082**	**0.8088**	**0.9322**	**0.9486**
ETC_ENTITY	**0.814**	**0.9098**	**0.9114**	**0.809**	**0.9326**	**0.9489**
ETC_RESTR_NOENT	**0.8152**	**0.9137**	**0.9154**	**0.8131**	**0.936**	**0.9513**

Readability was estimated as mean average (MA) scores per summary over relevancy(T), soundness (no unresolved anaphora) (A), non-redundancy (R) and syntactical correctness (S) among relevant passages of the ten tweets having the largest text references. The score of a summary was the average normalized number of words in valid passages. Sentence ordering was not judged by conference organizers.

In 2013 according to all metrics except redundancy our approach was the best among all participants (see Table 3). Runs were officially ranked according to mean average scores. Readability evaluation also showed that the run 275 is the best by relevance, soundness and syntax. However, the run 274 is much better in terms of avoiding redundant information. The runs 273 and 274 are close according readability assessment as well.

In 2014 in general the informativeness results were opposite to readability ones. However, our runs kept the same relative order. We received very low score for diversity and structure. This may be related to the fact that we decide not to treat this problem since in previous years their impact was small. Despite we retrieved the entire sentences from the Wikipedia, unexpectedly we received quite low score for syntactical correctness.

5.2 Snippet Retrieval

Evaluation was performed manually by the organizers. The goal is to determine the effectiveness of a snippet to provide sufficient information about the corresponding document. To this end, assessors should evaluate results in two ways:

- relevance evaluation of documents;
- relevance evaluation of snippets.

Table 3. Readability evaluation 2013

Rank	Run	MA	T	R	A	S
1	275	72.44%	76.64%	67.30%	74.52%	75.50%
2	274	71.71%	74.66%	68.84%	71.78%	74.50%
3	273	71.35%	75.52%	67.88%	71.20%	74.96%

The runs were evaluated by the following measures: Mean prediction accuracy (MPA), Mean normalized prediction accuracy (MNPA), Recall, Negative recall (NR), Positive agreement (PA), Negative agreement (NA), and Geometric mean (GM) of recall and negative recall. The official ranking was based on GM. The results for Snippet Retrieval Track are given in the Table 4. Our approach demonstrated the highest performance. As we hypothesized the knapsack algorithm provided better results since it searches for the most valuable information regardless its position.

Table 4. Snippet evaluation 2013

Run	MPA	MNPA	Recall	NR	PA	NA	GM
knapsack	**0.8300**	**0.6834**	**0.4190**	**0.9477**	**0.4921**	**0.8673**	**0.5352**
MW	**0.8300**	**0.6459**	**0.3852**	**0.9067**	**0.4283**	**0.8572**	**0.4605**
Focused_Split	0.8214	0.6549	0.3684	0.9413	0.4358	0.8624	0.4732
Focused	0.8171	0.6603	0.3507	0.9700	0.4210	0.8675	0.4774
Baseline	0.8171	0.6414	0.2864	0.9964	0.3622	0.8711	0.4025

References

1. Amini, M.R., Tombros, A., Usunier, N., Lalmas, M.: Learning–based summarisation of XML documents. Inf. Retr. 10(3), 233–255 (2007)
2. Arora, R., Ravindran, B.: Latent dirichlet allocation based multi–document summarization. In: Proceedings of the second workshop on Analytics for noisy unstructured text data. pp. 91–97. ACM, Singapore (2008)

3. Asher, N., Lascarides, A.: Logics of Conversation. Cambridge University Press (2003)
4. Barzilay, R., Elhadad, N., McKeown, K.R.: Inferring strategies for sentence ordering in multidocument news summarization. Journal of Artificial Intelligence Research pp. 35–55 (2002), 17
5. Barzilay, R., McKeown, K.R., Elhadad, M.: Information fusion in the context of multi–document summarization. ACL '99 Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics pp. 550–557 (1999)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research pp. 993–1022 (2003), 3
7. Carpineto, C., Romano, G.: A survey of automatic query expansion in information retrieval. ACM Computing Surveys 44(1), 150 (Jan 2012), http://dl.acm.org/citation.cfm?id=2071389.2071390
8. Delort, J.Y., Bouchon-Meunier, B., Rifqi, M.: Enhanced web document summarization using hyperlinks. In: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia. pp. 208–215. ACM, Nottingham, UK (2003)
9. Erkan, G., Radev, D.R.: LexRank: graph–based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research 22, 457–479 (2004)
10. Ermakova, L., Mothe, J.: IRIT at INEX: question answering task. In: Focused Retrieval of Content and Structure. vol. 7424, pp. 219–226 (2012)
11. Herndvlgyi, I.T.: Solving the sequential ordering problem with automatically generated lower bounds. Proceedings of Operations Research 2003 pp. 355–362 (2003)
12. Hovy, E., Tratz, S.: Summarization evaluation using transformed basic elements. Proceedings TAC 2008 (2008)
13. Landauer, T.K., Foltz, P.W., Laham, D.: Introduction to latent semantic analysis. Discourse Processes pp. 259–284 (1998), 25
14. Lapata, M.: Probabilistic text structuring: Experiments with sentence ordering. Proceedings of ACL pp. 542–552 (2003)
15. Lin, C.Y.: Assembly of topic extraction modules in SUMMARIST. In AAAI Spring Symposium on Intelligent Text Summarisation pp. 53–59 (1998)
16. Lin, C.Y., Hovy, E.: Identifying topics by position. Proceedings of the fifth conference on Applied natural language processing pp. 283–290 (1997)
17. Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of Research and Development pp. 159–165 (1958)
18. Meij, E., Weerkamp, W., de Rijke, M.: Adding semantics to microblog posts. In: Proceedings of the fifth ACM international conference on Web search and data mining. p. 563572. WSDM '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2124295.2124364
19. Milne, D., Witten, I.H.: An effective, low–cost measure of semantic relatedness obtained from wikipedia links. Proceedings of AAAI pp. 25–30 (2008)
20. Murdock, V.G.: Aspects of sentence retrieval. Dissertation (2006)
21. Niemann, E., Gurevych, I.: The peoples web meets linguistic knowledge: Automatic sense alignment of wikipedia and WordNet. International Conference on Computational Semantics (2011)
22. Paul, M.J., Zhai, C., Girju, R.: Summarizing contrastive viewpoints in opinionated text. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 66–76. Association for Computational Linguistics, Cambridge, Massachusetts (2010)

23. Radev, D.R., McKeown, K.R.: Generating natural language summaries from multiple on–line sources. Computational Linguistics – Special issue on natural language generation 24(3), 469–500 (1998)
24. SanJuan, E., Moriceau, V., Tannier, X., Bellot, P., Mothe, J.: Overview of the INEX 2011 question answering track (QA@INEX). In: Geva, S., Kamps, J., Schenkel, R. (eds.) Focused Retrieval of Content and Structure, Lecture Notes in Computer Science, vol. 7424, pp. 188–206. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978--3--642--35734--3_17
25. Seki, Y.: Automatic summarization focusing on document genre and text structure. ACM SIGIR Forum 39(1), 65–67 (2005)
26. Shen, D., Sun, J.T., Li, H., Yang, Q., Chen, Z.: Document summarization using conditional random fields. In: Proceedings of the 20th international joint conference on Artifical intelligence. pp. 2862–2867. Morgan Kaufmann Publishers Inc., Hyderabad, India (2007)
27. Sun, J.T., Shen, D., Zeng, H.J., Yang, Q., Lu, Y., Chen, Z.: Web–page summarization using clickthrough data. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 194–201. ACM, Salvador, Brazil (2005)
28. Teufel, S., Moens, M.: Sentence extraction and rhetorical classification for flexible abstracts. In Intelligent Text Summarization pp. 16–25 (1998)
29. Teufel, S., Moens, M.: Summarizing scientific articles: experiments with relevance and rhetorical status. Computational Linguistics 28(4), 409–445 (2002)
30. Wan, X., Yang, J., Xiao, J.: Manifold–ranking based topic–focused multi–document summarization. Proceedings of the 20th international joint conference on Artifical intelligence pp. 2903–2908 (2007)
31. Wang, D., Li, T., Ding, C.: Weighted feature subset non–negative matrix factorization and its applications to document understanding. In: Proceedings of the 2010 IEEE International Conference on Data Mining. pp. 541–550. IEEE Computer Society (2010)
32. Xu, J., Croft, W.B.: Improving the effectiveness of information retrieval with local context analysis. ACM Trans. Inf. Syst. 18(1), 79–112 (Jan 2000), http://doi.acm.org/10.1145/333135.333138
33. Yang, Z., Cai, K., Tang, J., Zhang, L., Su, Z., Li, J.: Social context summarization. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. pp. 255–264. ACM, Beijing, China (2011)