Incremental Summarization using Taxonomy

DongHyun Choi
Computer Science Department
Semantic Web Research Center, KAIST
cdh4696@world.kaist.ac.kr

Key-Sun Choi
Computer Science Department
Semantic Web Research Center, KAIST
kschoi@cs.kaist.ac.kr

ABSTRACT
In this paper, a new summarization system is proposed, which summarizes a document by interactively scoring the sentences using already-extracted summary so that the sentence which contains the most amount of relevant information to the summary to be extracted. To calculate the amount of relevant information contained by a sentence, the system makes heavy use of already existing taxonomies. The experiment shows a result pretty close to the current state-of-the-art systems.

Categories and Subject Descriptors
I.2.6 Knowledge Acquisition

General Terms
Algorithms

Keywords
Summarization, Taxonomy

INTRODUCTION
Text summarization is the process of automatically creating a short summary, from a document or a set of documents. Since automatic text summarization has many applications, e.g. making documents easily readable from small display panel, decreasing time and effort on understanding documents, the research on text summarization has drawn lots of attention from many researchers.

[1] claimed that, in the summarization process done by human experts, they extract relevant information on the purpose, scope, methods, results, and conclusions of the document, and then revise the document to ensure that the most amount of relevant information has in fact been packed into the compression. In this paper, we tried to use this analysis so that we can create more human-like summarization. More precisely, we suggest the method of interactive incremental sentence extraction, which extracts a sentence which supports the already-existing summary at most. We use the term “supportedness” to represent the amount of relevant information of a sentence to already-existing summary. Then, the system interactively and incrementally chooses the sentence whose supportedness is the largest, and then re-calculates the supportedness of each sentence to the new summary.

To calculate the amount of relevant information, the system depends heavily on the taxonomy. For example, consider the following two sentences:

(1) John is an animal.
(2) John is a dog.

Obviously, sentence (2) contains more information than sentence (1). That’s because ‘dog’ is also ‘animal’ – in other words, ‘animal’ is a taxonomical parent of ‘dog’.

So, the concept at the bottom of taxonomy contains more information than the concept at the top of taxonomy. The system numerically calculates the amount of information using the already existing taxonomies.

SYSTEM ARCHITECTURE
The summarization process done by the system is as follows: first, get the most important content of the document. Second, get the sentence with the most amount of relevant information to the most important content. Third, get the sentence with the most amount of relevant information to those already included in the summary, and include the sentence into the summary. The system iterates this process until every sentences are included inside the summary. The overall system architecture is described in figure 1.

Getting Supportedness
The supportedness value of a sentence is defined as the combination of two parameters: Direct Supportedness and Position Score.

Direct Supportedness
This feature represents how much a sentence supports the already-extracted summary, based on only the contents of the sentence and summary. The direct supportedness can be again divided as the combination of two components: Content Overlap. This feature represents the amount of summary content supported by this sentence.
**Added Information.** This feature represents the amount of newly introduced information by this sentence. Content overlap represents the number of keywords which are shared between the summary and sentence. The keywords are defined as the top \( n \) nouns with the highest TF values.

To get the added information, we should first calculate the information amount of each word in the sentence. The system uses Pointwise entropy, \( H(w) \), to do this:

\[
H(w) = -\log_2 p(w)
\]

where \( p(w) \) represents the probability of word \( w \) in \( n \)-gram model.

But we cannot simply apply the pointwise entropy to get the information amount for each word: For example, the word “entity” shows very rarely in real-world context, so the word “entity” will have very high pointwise entropy value – but we know that, the word “entity” could be everything, containing very small amount of information.

To solve this problem, taxonomies are used. More precisely, when constructing the language model, to get a probability of one word we count the emergence of the word plus the words below it in the taxonomy tree. For example, if a word “bus” is observed, than we think that a word “car” is also observed.

So the probability \( p' \) of word \( w \) in the modified language model is as follows:

\[
p'(w) = \sum_{w' \in D(w,T)} p(w')
\]

where \( D(w,T) \) represents the descendants of word \( w \) (including itself) inside the taxonomy \( T \).

Now, we can directly define the pointwise entropy of this new language model as the value of information amount for single word, \( I(w) \):

\[
I(w) = -\log_2 p'(w)
\]

The added information by sentence \( s \), \( AI(s, e) \), could be represented as the normalized sum of information amount of all single nouns, except those overlapped by already-existing summary. The information amount of single word is normalized to the value between 0 and 1.

Total direct supportedness of a sentence \( s \) given summary \( e \), \( DS(s, e) \), is defined as follows:

\[
DS(s, e) = CO(s, e) \cdot AI(s, e)
\]

where \( CO(s, e) \) represents the content overlap of sentence \( s \) and summary \( e \). We assumed that in one sentence, all of the newly introduced information is relevant to the contents included in the sentence.

**Position Score**

This feature represents the property that, if two sentences are positioned closely, then it is highly probable for them to support each other. In this system, the position score of a sentence \( s \), \( PS(s, e, d) \), is the function of sentence distance between the first selected sentence and \( s \):

\[
PS(s, e, d) = \frac{n - \text{Distance}(s, l, d)}{n}
\]

where \( l \) is the last sentence of the summary \( e \), \( n \) represents the number of sentences in the given document, \( d \) represents the input document, and \( \text{Distance}(s, l, d) \) represents the number of sentences between sentence \( s \) and sentence \( l \) at the input document \( d \). \( PS(s, e, d) \) increases when the position of a sentence is close to the last-extracted sentence, and decreases otherwise.

**Calculating Supportedness Value**

Supportedness value of a sentence \( s \), given summary \( e \), is defined as the weighted multiple of two parameters \( DS(s, e) \) and \( PS(s, e, d) \):

\[
SV(s, e) = DS(s, e)^\alpha \cdot PS(s, e, d)^\beta
\]

**EXPERIMENTS**

The experiments are done in various \( \alpha \) and \( \beta \) value settings. For the test corpus, we used 567 English news articles and their summaries provided by Document Understanding Conference 2002 [2]. For the evaluation, ROUGE-1 and ROUGE-2 metrics are used.

In the below table, \( S1 \) represents our system with \( \alpha = 1, \beta = 5.5 \) with both directed supportedness and position score as system feature, where \( S2 \) represents [3]'s system, and \( S3 \) represents [4]'s report on their system.

**Table 1. Comparison of System Results**

<table>
<thead>
<tr>
<th>System</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>49.34</td>
<td>25.80</td>
</tr>
<tr>
<td>S2</td>
<td>49.22</td>
<td>25.68</td>
</tr>
<tr>
<td>S3</td>
<td>42.02-50.23</td>
<td>13.61-25.47</td>
</tr>
</tbody>
</table>

**ACKNOWLEDGMENTS**

This work was supported in part by MKE & IITA through IT Leading R&D Support Project.

**REFERENCES**


