“How-to” Questions Answering Using Relations-based Summarization

Mikhail Aleksandrovich Dykov, Pavel Nikolaevich Vorobkalov and Valery Anatolyevich Kamaev

Volgograd State Technical University, Volgograd, Russia

Submitted: Aug 10, 2013; Accepted: Sep 21, 2013; Published: Sep 25, 2013

Abstract: The problem considered in this paper relates to searching for "How-to" question answers and identifying the main semantic part in the answers found. We propose a “bag-of-relations” method for document summarization. This approach consists in identifying sentences that correspond to the key relations the most. Rating of each relation in the training set of question answering documents is evaluated on the basis of its frequency, as well as on the basis of semantic closeness to other relations included in the set. We propose a method allowing automatic selection of documents for the training set and for refining the results of a full-text search. We also suggest a method of evaluating the quality of the document containing an answer to a "How-to" question, as well as a method of identifying main objects required to perform the actions described in the document found.

Key words: NLP • Summarization • Question answering • Bag-of-relations • Semantics

INTRODUCTION

Information is one of the most valuable assets in modern society and we need to provide user a way to access this large amount of information. We need systems that allow user to ask a question in everyday language and get an answer. Today existing search engines can return ranked lists of documents, where the user has to find the answer by himself. This problem can be solved with question answering systems. Question answering is a form of information retrieval that deals with natural language questions. The main task is to retrieve explicit answers to user’s questions [1].

The simplest form of question answering is dealing with factoid questions, for example: “When was Newton born?”, “What is the capital of Great Britain?”, e.t.c. Answers to such questions are simple facts that can be found in short text strings: a name of a person, a temporal expression, or a location. There are a lot of existing effective techniques for answering these kinds of questions. [2, 3]. Sometimes the answer to the question is not a simple fact extracted from the document. In such cases we might need a summary of a document or set of documents. The goal of text summarization is to retrieve the most important information from a source (or sources) to produce an abridged version for a particular user (or user) and task (or tasks) [4].

Text summarization can be divided into generic and query-based summarization. The first type is a way to represent a text as a set of sentences of phrases that contain main features of a source text. Query-based approach is a way to summarize a document with respect to the information given in a user query [5].

Summarization can be single- and multi-document. In the first case we need to produce an abstract from a single document. Multi-document summarization is a process of generating a generic or topic-focused summary by reducing documents in size while retaining main characteristics of the original documents [6].

Summarization can also be divided into extractive and abstractive. Abstraction involves reformulation of contents, while in extraction method the most important sentences of the original document are picked up for summary generation [7].

"How-to" questions include common questions such as "How to paint a wall?", "How to fix a pipe leak?", or "How to make a cake?". Answers to these questions contain sequences of actions leading to results. Some of these actions can be optional, obvious or be a part of detailed description of other actions. Thus, the entire text answers the question and consequently, generic summarization is needed to identify the semantic part. In our approach we use an extractive single-document summarization. Extractive summarization requires ranking sentences according to their importance.
There are several methods of determining sentence importance.

Traditional method is based on evaluation of term and inverse document frequencies (tf*idf) [8].

The main idea of Graph-based methods is that sentences that are similar to many of the other sentences in the document are more important. One of the measures to evaluate the strength of relationship among the sentences is cosine metric [9].

\[
\text{Cosine}(x, y) = \frac{\sum_{w \in x, y} tf_{w,x} \cdot f_{w,y} \cdot (idf_w)^2}{\sqrt{\sum_{w \in x} (tf_{w,x} \cdot idf_w)^2} \cdot \sqrt{\sum_{w \in y} (f_{w,y} \cdot idf_w)^2}}
\]

Both types of the methods mostly use words or phrases (n-grams) as a metric for evaluation of (tf*idf). The drawback of unigram models is that they only calculate the probability of hitting an isolated word, without considering any influence from the words before or after the target. Bigrams require a big training set. N-gram approaches also fail to account for the semantic similarity among words in different sentences, inferring some information loss during searching for the best sentences.

There are also some approaches based on knowledge bases. These approaches allow to capture the semantic interrelations within the text and determine the crucial information based on these relations. But disadvantages of these approaches are their domain dependence and knowledge intensity [10].

**MATERIALS AND METHODS**

In order to overcome the restriction of the considered methods, we developed the bag-of-relations approach to identify the main semantic part of the text. Under the term "relation" we understand grammatical relation - the edge in the Stanford dependency tree, the syntactic relationship between two words present in a clause: the head and its dependent. For example, the sentence: "She gave me a raise" contains grammatical relation "(gave, raise)". The type of this relation is "direct object". The direct object of a verb phrase is the noun phrase which is the object (accusative) of the verb [11].

This approach requires a training set of documents containing answers to the "How-to" question asked. We decided to select documents for this sample from the search results presented by the existing search systems.

Our method allows identifying the semantic part from hypertext documents, which contain an answer to the question asked, as well as separating main information from summaries and lists of required objects. The application of this method makes it possible to create a training set for practically any common "How-to" question.

Our approach includes identifying parts of speech for all words in the documents and subsequent indexing of words belonging to the most meaningful parts of speech: nouns, verbs and adjectives. By indexing only those words that are present in several documents at the same time with their further normalization based on the overall frequency of use of these words based on the data from special frequency dictionaries [12], it is possible to single out concepts which are most characteristic of the subject area under consideration.

In our approach we single out only those relations that contain the identified key concepts. We suggest a method for indexing relations, which is based on determining the frequency of their entry to the training set, as well as on the semantic closeness of the relations. The application of this approach makes it possible to identify the main steps to do in the question answer.

Our method of determining text relevance to the key relations of the subject area makes it possible to evaluate the quality of the documents containing answers to the question asked, as well as to evaluate the importance of each of the sentences in the text considered for the answer to the "How-to" question.

We also propose a method based on the indexed key relations and concepts, which allows identifying blocks of short step-by-step answers to the question asked from hypertext documents, as well as blocks of objects required for solving the question.

**Input and Preprocessing:** To compile the relations index, we need a set of web documents \( S = \{WD_1, WD_2, WD_n\} \) containing the answers which are the most relevant to the "How-to" question. One can obtain this sample from search results of web search engines. These systems have their own mechanisms for ranking web documents and, therefore, the documents placed at the top of the search results can be considered as relevant to the question asked. The web documents containing the answer to the "How-to" question can consist of four parts: \( WD = (D, S, I, M) \) where \( D \) is the text part of the document which contains the answer, \( S \) is a short step-by-step answer to the questions, \( I \) refers to objects required to solve the "How-to" problem and \( M \) is a
document markup and html tags. Components \( S \) and \( I \) may not be present. In our case, to identify the semantic part of the web document \((D,S,I)\), we need to maximize recall, since the main point is not to miss information which is needed to answer the question, whereas the entire text included in \( M \) is, as a rule, unique for each web document and screened out during indexing.

In the web document, \( D \) is a text consisting of a sequence of a few sentences. Its identification takes place through a regular expression identifying the sentence in the text block.

The resulting regular expression gives recall = 0.98 and precision = 0.9 in average on the set of 100 web pages containing answers.

In 90% of web documents, \( S \) and \( I \) are list elements, each item of which is surrounded by \(<li>\) html tags. The attribution of the identified lists to \( S \), \( I \) or \( M \) is based on the relations index and key concepts of the subject area.

Let us define the set of key concepts of the subject area as:

\[
C = \{ (c_1, \eta_1), (c_2, \eta_2), \ldots, (c_m, \eta_m) \}
\]

\[
c_i \in \{ D_j \mid D_{j+1}, \ldots, D_{j+d} \}, \eta_i \geq 0.15 \cdot n
\]

\[
c = \{ w, p, sen \}
\]

\[p \in \{ \text{noun, verb, adjective} \}
\]

where \( c_i \) is a concept, \( r_i \) is the word rating: normalized number of word entries to \( \{ D_j, D_{j+1}, \ldots, D_{j+d} \} \) \( w \) is a word, \( p \) is a part of speech, \( sen \) is a sense, \( n \) - the number of documents, \( m \) - the total number of different concepts. Thus, most high-frequency words which are not characteristic of the given subject area, will not be included in \( C \). To minimize the weight of the remaining high-frequency words which are not related to the subject area, we perform a normalization of \( r_i \) based on the unigram frequency dictionary. In the answers to "How-to" questions the main parts are actions, objects to perform this actions and properties of these objects. So we limited \( c_i \) as a noun, verb, or adjective. To identify parts of speech in each \( \{ D_j, D_{j+1}, \ldots, D_{j+d} \} \) we used the Stanford parser [13]. The marked parts of speech will also be used later for finding semantic closeness of words using WordNet [14].

Relations Indexing: Let us define the set of key relations of the subject area as:

\[
RL = \{ (r_{i_1}, r_{i_1}), (r_{i_2}, r_{i_2}), \ldots, (r_{i_n}, r_{i_n}) \}
\]

where \( r_{i_1} \) is the relation, \( r_{i_1} \) is the relation rating in \( \{ D_j, D_{j+1}, \ldots, D_{j+d} \} \).

\[
rl = \{ RT, O_1, O_2 \}
\]

where \( RT \) is the relation type, \( O_1, O_2 \) - are objects(words) included in the relations. In the answers to "How-to" questions, the important things are the actions, the items which are used to perform the actions or which the actions are performed on, as well as the properties of the items. That is why the following relation types are the key: dependent, direct object, clausal complement with external subject, adjective modifier. Thus,

\[
RT \in \{ \text{dep, conj, dobj, xcomp, amod} \}
\]

where \( \text{dep} \)- dependent relation, \( \text{conj} \)- conjunct relation, \( \text{dobj} \)- direct object relation, \( \text{xcomp} \)- clausal complement with external subject relation, \( \text{amod} \)- adjective modifier relation.

The relations are singled out by means of the Stanford parser. Relation is considered if \((O_1 \in C)\) and \((O_2 \in C)\).

To determine the \( r_{i_1} \) relation rating, we developed the following method:

\[
r_{i_1} = a_i + \sum_{j=1, j \neq i}^{l} k_{ji} \cdot a_j
\]

where \( a_i \) is the number of entries of \( r_{i_1} \) relations to \( \{ D_j, D_{j+1}, \ldots, D_{j+d} \} \), \( k_{ji} \) is the semantic closeness ration between \( r_{i_1} \) and \( r_{j_1} \) relations.

\[
k_{ji} = \begin{cases} 
0, & \text{if } RT_i \neq RT_j \cup \left( \text{synonym} \{ O_1, O_j \} \cap \text{hyperonym} \{ O_1, O_j \} \right) \\
\frac{1}{d(O_1, O_j) + d(O_2, O_2)} & \text{otherwise}
\end{cases}
\]

where \( k_{ji} \) is the distance between concepts in WordNet (Miller, Beckwith, Fellbaum, Gross and Miller 1993).

To minimize the time needed for finding semantic closeness for each word pair, we indexed all possible synonyms and hyperonyms for all \( \{ O_1, O_2, \ldots, O_N \} \).

\[
In = \{ In_1, In_2, \ldots, In_{MM} \}
\]

\[
In_i = \left\{ \frac{1}{d(O_1, O_j)} \bigg| d(O_1, O_j) \right\}
\]

\[
synonym(O_1, O_j) \cap hyperonym(O_1, O_j) \cap O_j = O_i
\]
where $In$ is the index of synonyms and hyperonyms.

Thus, for identifying words $\{O_1, O_{i+1}, O_{i+k}\}$ that have a certain semantic closeness with word $Q$ and distances $d$ to these words, it is necessary to find $b_m = O_j$ and single out all links related to $in_m$ from the index.

Main texts ranking and short test blocks classification

The quality of the document is evaluated using the following method:

$$Q = \frac{\sum_{i=1}^{n} r_i \cdot a_i \cdot RT_i \in RT}{\sqrt{O_{amount}}} \cdot \sqrt{\frac{nu}{O_{amount}}}$$  \hspace{1cm} (9)

where $n$ is the number of different relations that occur in the document, $r_i$ is the $i$-relation rating, $RT_i$ is the relation type, $RT$ is the set of all allowed relations, $a_i$ is the number of $i$-relation in the text, $O_{amount}$ is the number of relations in the relations index and $nu$ is the number of different relations from the relations index, which occur in the document. Thus, the quality of the text is determined by means of two constituents:

- Constituent that characterizes the average rating of all relations in the documents. This constituent characterizes the degree of relevance of the document to the subject area.
- Constituent that characterizes the level of coverage of the subject area by the document.

Let us come back to the other blocks singled out from $WD$: $(S,I,M)$. Let us assume that:

$$U \in \{S, I, M\}$$  \hspace{1cm} (10)

where $U$ is a block of unknown type.

Let us calculate $Q(U)$ and $Q_s(U)$, where

$$Q_s(U) = \frac{\sum_{i=1}^{m} r_{c_i} \cdot ac_i \cdot p_i \in p}{\sqrt{C_{amount}}} \cdot \sqrt{\frac{muc}{C_{amount}}}$$  \hspace{1cm} (11)

$$Q(U) = \frac{\sum_{i=1}^{m} r_{c_i} \cdot ac_i \cdot p_i \in p}{\sqrt{O_{amount}}} \cdot \sqrt{\frac{muc}{O_{amount}}}$$

where $m$ is the number of different concepts which occur in the document, $rc_i$ is the $i$-concept rating, $ac_i$ is the number of $i$-concepts in the text, $C_{amount}$ is the number of concepts in the set of key concepts of the subject area, $muc$ is the number of different key concepts of the subject area which occur in the document, $p_i$ is the part of speech of the word, $p$ is the set of allowed parts of speech.

$$\begin{align*}
Q(U) > b_1, Q_s(U) > b_2 & \quad U = S \\
Q(U) > b_1, Q_s(U) < b_2 & \quad U = I \\
Q(U) < b_1, Q_s(U) < b_2 & \quad U = M
\end{align*}$$  \hspace{1cm} (12)

where $b_1$ and $b_2$ are certain boundary values.

In this way, the summary and main object blocks used for performing actions described in the document are identified in the web document.

**Main Semantic Part Selection:** To evaluate the quality of each sentence $sen$ included in the document, we suggest the following method:

$$Q_{sen} = \frac{\sum_{i=1}^{m} r_{c_i} \cdot RT_i \in RT}{m}$$  \hspace{1cm} (13)

where $m$ is the number of different relations that occur in the sentence, $r_i$ is the $i$-relation rating, $RT_i$ is the relation type, $RT$ is the set of all allowed relations types.

Then, we evaluate the quantity of sentences that need to be included in the final summary:

$$Sencount = \min_{i=1}^{t} \left( scount(S_i), Q(S_i) > b \right)$$  \hspace{1cm} (14)

where $t$ is the number of short step-by-step answers $S$ in the sample of web documents $WB$, $scount(x)$ is the function for calculating the number of sentences in the text, $Q(S)$ is the quality of the summary, $b$ is a boundary value which determines the minimal quality of the short answer.

Thus, the number of sentences that need to be included in the final summary of the document will correspond to the number of sentences in the shortest quality step-by-step answer singled out from the training set. The final summary of the document can be presented as follows:

$$Sum = \{\{Sen_1, Sen_2, ..., Sen_{Sencount}\} \cap I_1 \cap I_2 \cap I_o\}$$  \hspace{1cm} (15)

Experiments: To test the suggested methods, we chose 10 "How-to" questions, such as:

- How to hang wallpaper?
- How to paint a wall?
- How to mount a TV?

For indexing relations and key concepts for each of the questions, we selected 5 random documents out of 10 top documents in search engine results. The remaining five documents were included in the test sample. We used certain filters to ignore documents containing video
Fig. 5: Results of scoring summaries and short step-by-step answers

answers. We also took 5 documents from other subject areas. Then we evaluated their relevance using formula (9), Unigram and Bigram methods and manually by experts. We trained Unigram and Bigram models on the same training set, considering only the words for the same parts of speech as the ones used for training in our approach. The results for one question are given in Figure 1, Figure 2 and Figure 3. All ratings are normalized by the highest rating.

Then, in the best answer to each 'how-to' question, we identified the main semantic part using the suggested method, as well as unigram and bigram methods. The examples of the generated summaries are given in Figure 4.

Then, we performed an assessment of the quality of the identified main semantic parts (Sum), as well as the quality of short answers (SList) to the "How-to" questions singled out from web documents. The assessment was performed using the formula (9) and manually. The results are provided on the Figure 5.

RESULTS AND DISCUSSION

As we can see from Figure 1, Figure 2 and Figure 3, the suggested bag-of-relations technique shows the nearest result to manual scoring. But, as we can see that Bigram approach shows quite similar results. The meaning of this is that in this experiment we used for Bigrams only parts of speech that we used for relations and the words in many relations are located next to each other. However, Figure 4 shows that our approach finally generates better summary. We also can see that all documents from the test set have lower average score than documents from the training set. It happens because of the small training set. A larger training set can't be obtained because of the limitation of high relevance documents from the top of search engine results and time that should be spent for training. Mostly it influences results of applying Bigrams and marginally on Unigrams. It happens because in the case of small number of training examples Bigrams and Unigrams cannot detect the semantic similarity between words in the training set and new words in the test set. This can be done by using relations and by calculation the semantic similarity between them. In our case for the ultimate goal of the summary generation we can use documents from the training set. All techniques separate documents from different categories. But as we can see bag-of-relations and Bigrams methods still give a higher score to this document compared with Unigrams. The reason of this is that the document contains much more "weighable" Unigrams than relations or Bigrams and even a single relation or Bigram has a big impact on the document score.

Figure 5 shows that generated summaries get good but still worse scores than hand written short step-by-step answers. Our scoring method also gives higher score for these answers. The reason is that short step-by-step answers consist of high relevant dependent relations and summaries are made from whole sentences of the complete answer. The low quality summary for the document number 4 occurs because it was a document with multiple answers on the question and the final summary included sentences from both answers. This problem can be solved by applying the graph-based approach for evaluating the strength of relationship between sentences [15]. The document number 5 contained a short high quality solution for "How-to" question that wasn't recognized as short step-by-step answer. That's the way it was scored higher than the short step-by-step answer from another document.

The developed method can be improved in the following ways:

- Improving the POS tagger, as it gives F1 = 84.2. Experiments with handmade POS tagger promise to improve results;
- Improving the relations parser, as not all meaningful relations are singled out in complex sentences;
- Using a sense determination method before searching for synonyms and hyperonyms in WordNet;
- Taking into consideration not only direct relations between words, but also relations between relations, i.e. considering more levels of hierarchy in the dependency tree;
**Conclusion and Future Work:** We have suggested an approach to searching for answers to "How-to" questions and identifying their main semantic part. Using the semantic part identification method for web documents, we have obtained training sets required for the suggested document quality evaluation based on the bag-of-relations. The experiments showed the effectiveness of the suggested approach in comparison with the methods based on unigrams and bigrams.

The developed methods can also be used for other purposes. Our document scoring methods can be used to ensure more relevant search engine results for queries classified as the "How-to" type. The method of calculating relations semantic similarity can be used for the trend detection in Social Media [16]. Short summaries for “How-to questions answers” can be used for automatization of everyday creativity [17].

In the future, we plan to develop a method for compiling a summary for answers to "How-to" questions based on several documents. We also want to extend our approach for other types of questions.

**REFERENCES**