Semantic Analysis for Document Similarity and Search Queries

Charlene Cassar and Samad Ahmadi

Abstract Document similarity is a popular topic in natural language processing fields, especially in areas of Information Retrieval. Current systems are often limited to keyword search due to the complexities of handling free text search. Accurate results are rarely achieved. In this research we propose a combination of different semantic analysis methods for using with free text searches. Our aim is to enable the user to describe their search more freely. We will be using semantic analysis tools to understand the context of search and find more relevant results. Amongst several different domains which could be chosen for proof of concept, for our case study we have chosen extracting information from dream interpretation websites. In this case study our research aims to return results based on specific dreams input from users. This is part of an ongoing research and results so far are satisfactory.

1 Introduction

Most current search engines base their results on indexing keywords or expressions and are sensitive to the specific keywords used. The aim of this research is to test the hypothesis of whether or not the use of NLP techniques, such as keyword extraction and semantic analysis can be used to correct and enrich a search query in the form of text descriptions into relevant search results acceptable to humans. Our aim is to provide relevant results based on large amounts of text rather than keyword submissions. For proof of concept we have chosen dream interpretations web

C. Cassar (✉) · S. Ahmadi
Faculty of Technology, De Montfort University, Leicester, UK
e-mail: charlenecas@gmail.com

S. Ahmadi
e-mail: sahmadi@dmu.ac.uk

M. Bramer and M. Petridis (eds.), *Research and Development in Intelligent Systems XXXII*, DOI 10.1007/978-3-319-25032-8_7

sites and dictionaries where user queries naturally come as descriptions rather than keywords. In order to satisfactorily find interpretations for users descriptions of their dreams, steps are: to accept a piece of text depicting a dream, process this text and return a suitable list of interpretations.

2 Previous Research

Due to the nature of semantic analysis on search queries, we focus on query processing for a specific domain, dream interpretation. In the case of Question Answering (QA) a query must be submitted by a user, the query is then processed, and a large amount of documents are retrieved and ranked according to likeliness of containing the correct answer. A close similarity between QA field and dream interpretations can be seen where the user submit their dream (in the case of dream interpretation); this text is processed using NLP techniques; and from a list of interpretations (documents retrieved from a database), the similar documents to the query are returned as the interpretation.

For the scope of this work, the evaluation method will focus on the information retrieval metrics. The users would have the chance to enter a piece of text about their dream and rate the result. If they find that the documents returned are relevant to the query, the user will then give the system a positive rating; otherwise a negative rating is expected.

[2] mention two evaluation techniques which are useful to evaluate information retrieval which are known as the retrieval performance evaluation. Two of these popular techniques are known as the recall and precision evaluation methods.

Recall
The recall method measures the amount of documents that were retrieved in comparison to the amount of documents that were relevant and should have been retrieved. This can be defined as:

$$\text{Recall} = \frac{|\text{Ra}|}{|\text{R}|}$$

where Ra is the number of documents retrieved (which were relevant) and R is the total number of documents which should have been retrieved.

Precision
Precision is a measure of the amount of relevant documents retrieved in comparison to the total number of documents (regardless of whether they are relevant or not) retrieved. Precision is defined as:

$$\text{Precision} = \frac{|\text{Ra}|}{|\text{A}|}$$

Similar to the recall method mentioned above, Ra is the total number of relevant documents retrieved. A is the total number of documents retrieved.

For the scope of this research, precision and the recall metrics are used. The text input will be short mainly due to time constraints, since it would take a very long time to sift through all of the documents available to find the relevant documents which should have been retrieved. In this research ten different dream queries will be used for evaluation purposes [4].

In [5] the researchers divide the question answering task into two main stages; the syntactic analysis and the semantic analysis stage. The first stage takes the query and transforms this into an array of keywords. The semantic analysis obtains a semantic representation with the use of domain-specific ontologies to derive their relationships.

[7] concentrates on processing a user query (in the case of the current research, the dream submitted will be the query). Several Natural Language Processing techniques are used in the four layers which Pasca defines to be necessary.

In the first layer Pasca focuses on obtaining the lexical terms. Each term within the text is considered separate, bearing no relation to the other words in the textual query. The stemming algorithm is then applied to each individual term (Porter's Stemming Algorithm) to obtain the root word from which the term was derived [7].

Each term in the query is given a part-of-speech (POS) tag, which basically identifies the type of word (noun, verb, etc.). The position of the term within the query is also obtained. After this information has been extracted and the terms have been tagged, the terms are divided into two groups; content and non-content lexicons [7].

Content words are the terms which contribute to identifying a relevant document. Content words are usually tagged as nouns, verbs, adverbs or adjectives whilst non-content words are usually modal verbs, auxiliary verbs, prepositions, conjunctions, pronouns and interjections. For clarification purposes we will take an example; "The President of Malta was wearing a coat". The content words would be President, Malta, wearing and coat [7].

In the second layer the relationships between the content words are obtained. In the example given above we can easily see that there is a clear relationship between the terms President and Malta. This enabled Pasca to obtain a clearer idea of what the user required [7].

In the third layer, this will not be necessary in the case of dream interpretation, deals with the question stem and the expected answer type. Question stems are terms such as Who, What, How, Which, Why, Where, When Name, Whom, etc. Pasca found that How questions were amongst the most difficult questions to answer due to the ambiguity these questions pose. The expected answer type would be what answer type we expect to obtain from the question posed, for instance, person. However this layer does not related at all to the dream interpretation system described in this work [7].

In the fourth layer Pasca gives importance to semantic constraints. Semantic constraints occur when the meanings of words pose certain constraints due to the relationship between the terms for example, United Kingdom [7].

Once a significant number of keywords and their relationships have been obtained, the next step is to use information retrieval techniques to retrieve a set of relevant documents [7]. In the scope of this project these documents were the dream interpretations obtained.

For information retrieval, an inverted index may be used. The retrieved set of documents is taken, split into tokens (where a token is a word or phrase). The tokens are then stemmed to their normalized format. The inverted index is applied to a dictionary with a postings list and the dictionary contains the frequency of the term. The postings list is a list of all the documents containing the term [6].

In [2], the inverted index is also used to index relevant documents. The three most popular methods of Information Retrieval mentioned in [2] are the Boolean, Vector and Probabilistic methods.

The Boolean retrieval method tags documents as being relevant or not; relevant documents are tagged with a "1" and non-relevant documents are tagged with a "0" [1]. The Boolean method accepts user queries and matches these against a dictionary. For each query term, a postings list is obtained and an intersection is performed on these lists [2].

The Boolean Retrieval method is the most popular method of Information Retrieval since it can be simply represented formally and due to the simplicity of its use. This method is known to be the least likely to yield relevant results from the set than the other popular methods since the document can either be relevant or non-relevant without a rating for documents in between [2].

Due to the limitations of the Boolean Retrieval method, there is also the extended Boolean model, which caters for some of the original method's shortcomings. The extended Boolean retrieval method is found to retrieve more related documents than that of its ancestor [3]. This method is an extension of the Boolean method with the inclusion of the vector based method (described below) [2].

Another retrieval method is the Vector Space Model, which is also known as the cosine similarity measure [4]. The purpose of this method was to give a non-boolean (not limited to only 0 or 1) weight value to documents to give a clear indication of the degree to which the document is relevant.

This method also makes use of term frequency to acknowledge whether a document is relevant or not relevant to the query posed. The term frequency, which also goes by the name of tf factor, defines the document using terms as a feature. In other words, the tf factor defines the number of occurrences of that term within a particular document. The inverse document frequency uses a differentiating feature to define the document. This is the frequency rate of the term within the set of documents as a whole. If all of the documents contain a particular term, then the term can hardly be considered as a distinguishing feature [4].

After the query has been processed and the relevant documents have been retrieved, the user may then be presented with the results.

3 Dream Interpretations

Various psychologists such as Sigmund Freud and Carl Jung have based their careers on dream analysis [8]. Many people are interested in getting an insight to why they are experiencing certain dreams and the hidden meaning that may be behind these dreams. The current stance of the advancement of dream analysis interpreters is still quite premature.

Using natural language processing (NLP) we believe that a proper semantic analysis can be performed on the dream text input by the user and thus extract the meaning of the text (to differentiate from the dream interpretation, here we simply mean the semantic value of the text). From this, information retrieval algorithms are applied to the text sources retrieved from dream dictionaries and the highest matching documents are chosen to be the possible interpretation of the dream.

4 Semantic Analysis of Search Queries

This section describes details of our algorithms in our proposed system. The system accepts a user's text input and the language of the text is detected. If it is English it may proceed to the next stage of the process. The language of the text is detected using Google Translation API.[1]

The next step of the system is to ensure that the spelling of the text is correct. The NHunspell API[2] is used to check the spelling of the words and to suggest relevant recommendations when matched against its database. Due to time constraints, only a limited amount of slang word conversions were provided. The slang words are corrected when matching against the NHunspell database. Words such as "r" which is often used in place of the word "are" in slang are automatically converted.

To enrich the text with additional information, the OpenCalais API[3] is used. This library enables the semantic context of the text to be derived. These keywords would be added to the content word list derived from OpenCalais to increase the relevant documents found (Fig. 1).

In the next step, the lexical terms are derived, as stated in [7]. We used Porter's stemming algorithm to obtain the stem of the tokens in the text but realised that this gave us results such as "comput" instead of "computer" which might affect the retrieval of documents since they are indexed by their proper term "computer"

[1] https://cloud.google.com/translate/docs.

[2] http://www.crawler-lib.net/nhunspell.

[3] https://www.drupal.org/project/opencalais_api.

```
var openCalais = new Calais.CalaisDotNet(apiKey, content);
String doc = openCalais.Call<CalaisJsonDocument>().RawOutput.ToString();
List<String> tags = webScraper.GetSemantics(doc);
```

Fig. 1 Snippet of code calling OpenCalais API[4]

rather than their stem "comput". From this list of lexical terms, the content terms and non-content terms were identified. Part of this process involved automatically removing the stop words from the content words list. Stop words are terms which carry no semantic weight and thus do not change the meaning of the text, such as the words "a", "the" or "that", which definitely do not need to be used in the information retrieval process.

Once the content word list has been obtained, along with the additional information, the next step is to retrieve the documents. The interpretations for this system were obtained from the dreammoods website.[5] A process of webscraping was used in order to obtain the documents which relate to the content terms obtained. The page was parsed and the relevant interpretations were retrieved according to the queries set. Had there been more time available, a database could have been set up and the information could be loaded there. This would ensure less dependency on the website.

In more cases than not, there would be more documents retrieved than is necessary. A lot of these documents might simply contain one or two of these terms but in reality, have nothing to do with the dream inputted. Thus, the vector space model was chosen to handle this problem and only return the relevant documents to the user.

The vector space model was chosen because it considers the frequency of the occurrence of the terms in the documents. The term frequency and the inverse document frequency are calculated for each of the documents. Finally when this process is done the TF IDFs (multiplication of the TF value and the IDF value) is calculated. The last step of the vector space model is to apply the cosine similarity rule. A predefined threshold value is set and any document exceeding this value is added to the list and returned to the user [4].

If the documents were obtained from an unreliable source, they would need to be spell checked and have its slang corrected. However, since the documents are coming from a trusted website, we did not feel that this was necessary.

The following figure gives a representation of the algorithm flow for the semantic analysis of search queries (Fig. 2).

[4] http://calaisdotnet.codeplex.com/

[5] http://dreammoods.com/.

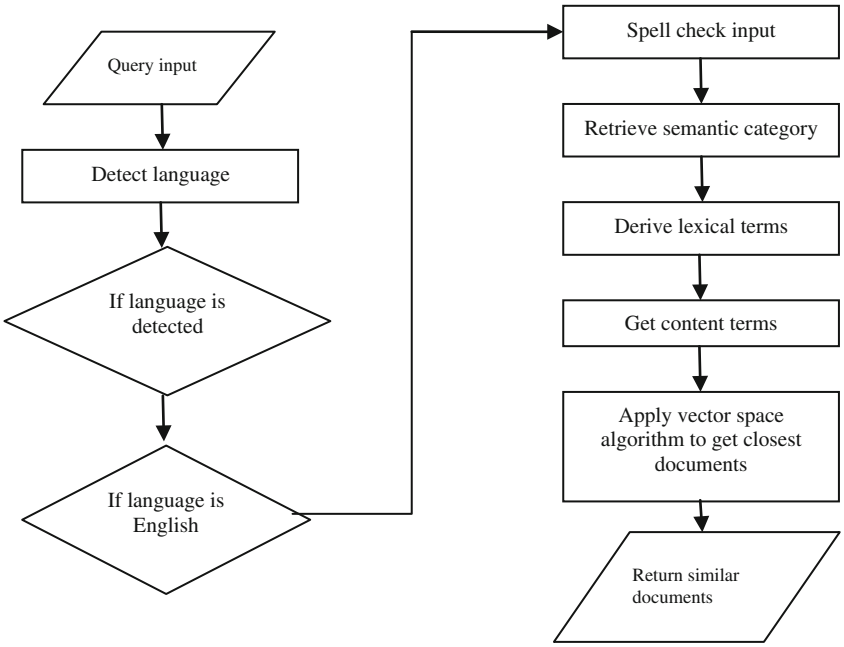

Fig. 2 Semantic analysis for search queries algorithm

5 Results

In most cases the program proved to yield relevant results but also had the tendency of returning more results than necessary.

For the purpose of the evaluation study, short but consistent dream queries are submitted to the system. This enables us to calculate the recall and precision rates in a reasonable time frame.

Initially the threshold was set to 0.15.

Precision

The results for the ten test cases can be seen below (Table 1):

The average precision rate for the ten documents was 0.3775; meaning 37.75 % of the documents retrieved by the system and returned to the user were relevant. The other 62.25 % were irrelevant.

Below we check for recall.

Recall

The results for the ten test cases can be seen below (Table 2):

The average recall rate for the 10 documents was higher than the precision rate. We managed to get a rate of 0.736; meaning that a percentage of 73.6 % of the documents that should have been retrieved, were in fact retrieved. The other 26.4 % of the documents which should have been returned were not returned to the user.

Table 1 Precision results for ten test cases with a threshold of 0.15

Test case	Relevant retrieved	Total retrieved	Precision rate
1	7	21	0.33
2	5	17	0.29
3	3	7	0.43
4	1	2	0.5
5	2	8	0.25
6	1	3	0.33
7	1	3	0.33
8	1	2	0.5
9	6	16	0.375
10	4	9	0.44

Table 2 Recall results for ten test cases with a threshold of 0.15

Test case	Relevant retrieved	Relevant should be retrieved	Recall rate
1	7	7	1
2	5	5	1
3	3	3	1
4	1	2	0.5
5	2	4	0.5
6	1	4	0.25
7	1	4	0.25
8	1	1	1
9	6	7	0.86
10	4	4	1

Next we altered the threshold rate to check if there was a difference in the results. The threshold was then set to 0.35.

Precision

The results for the ten test cases can be seen below (Table 3):

The average precision rate in this case was 0.305 (30.5 %). Considering that the precision rate was previously 0.3775, this value has decreased by increasing the threshold. That would mean that many relevant documents are ranked lowly (from between 0.15 and 0.35) and thus when the threshold is increased, these documents are removed.

Recall

The results for the ten test cases can be seen below (Table 4):

The total recall rate with a threshold of 0.35 was 0.445 (44.5 %), a dramatic decrease from the 73.6 % obtained with a threshold of 0.15.

As discussed above, the threshold was changed to a higher value, assuming that the precision rate would benefit, however the opposite effect was obtained. Less relevant documents were returned when the threshold was increased meaning that

Table 3 Precision results for ten test cases with a threshold of 0.35

Test case	Relevant retrieved	Total retrieved	Precision rate
1	4	11	0.36
2	5	17	0.29
3	2	6	0.33
4	1	2	0.5
5	1	7	0.14
6	1	5	0.2
7	0	1	0
8	0	1	0
9	5	9	0.56
10	2	3	0.67

Table 4 Recall results for ten test cases with a threshold of 0.35

Test case	Relevant retrieved	Relevant should be retrieved	Recall rate
1	4	7	0.57
2	5	5	1
3	2	3	0.67
4	1	2	0.5
5	1	4	0.25
6	1	4	0.25
7	0	4	0
8	0	1	0
9	5	7	0.71
10	2	4	0.5

many relevant documents were given weights between 0.15 and 0.35, thus omitted from the list when the threshold is 0.35. To obtain a better effect, the keywords fed to the information retrieval module must be enriched with much more information. It would be ideal if the document itself was parsed to ensure that it does not contain terms which have nothing to do with the dream inputted. This might eliminate irrelevant documents from being retrieved and increase the precision rate. The precision rate might increase if the input is of a longer length. To evaluate this, we submitted a dream found on the dreammoods[6] website:

> I have been having these nightmares about my girlfriend and kids being either attacked by 1 black dog; a shadowy figure, or the newest one from last night, they we're screaming for my help and I was running around looking for them franticly and couldn't find them. My girlfriend kept telling me the youngest one was hurt and they needed my help badly, and then she kept asking me why wouldn't I help them.

[6] http://dreammoods.com/.

For a threshold of 0.15 we have the following figures:

$$\text{Precision Rate} = 6/39 = 0.15$$

$$\text{Recall Rate} = 6/7 = 0.86$$

The precision rate obtained with a threshold of 0.35 was:

$$\text{Precision Rate} = 3/9 = 0.33$$

And the recall rate was calculated as follows:

$$\text{Recall Rate} = 3/7 = 0.43$$

As we can see above, a larger piece of text contributed to a higher recall rate when a smaller threshold was used but a lower threshold in both cases (smaller and larger threshold).

6 Conclusion and Further Work

From this research work we saw that question processing and information retrieval techniques can be used for dream interpretation systems. The accuracy of the system, which is still at a primitive level, depends heavily on the processing techniques.

The system did suffer a lag in time when it came to parsing longer pieces of text. In the case where the text was too long the program simply stopped functioning completely. This would definitely need to be fixed in the future. The aim would be to reduce the time complexity from O(n^2) to O(n), since various nested loops were used during processing.

As part of the further work, it would be interesting to experiment with the idea of query optimization using genetic algorithms to ensure that not only the correct documents are returned but also formulated to return the correct part of the document to the user.

In the future we intend to use databases to store interpretations, each of which will be indexed by its stem and provide additional information such as lexical terms and POS tagging. This will allow for a smoother mapping between query and document and would eliminate the dependency of the system on the dreammoods[7] website.

To summarize, we are satisfied by the result because it highlights the possibilities of using semantic analysis for document similarity and that such a system can exist.

[7] http://dreammoods.com/.

References

1. Azzopardi, J.: Template-Based Fact Finding on the World Wide Web. University of Malta, Msida (2004)
2. Baeza-Yates, R., Riberiro-Neto, B.: Modern Information Retrieval. ACM Press (1999)
3. Grundwls, L., Kwok, K.L.: Sentence Ranking Using Keywords and Meta-keywords. Springer, New York (2008)
4. Jurafsky, D., Martin, J.H.: Speech and Language Processing. Pearson Prentice Hall, New Jersey (2009)
5. Liang, J., Nguyen, T., Koperski, K., Marchisio, G.: Ontology-based natural language query processing for the biological domain. In: Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology, pp. 9–16. Association for Computational Linguistics, New York (2006)
6. Manning, C.D., Raghaven, P., Schutze, H.: An Introduction to Information Retrieval. Cambridge University Press, Cambridge (2009)
7. Pasca, M.: Open-Domain Question Answering from Large Text Collections. CSLI Publications, United States (2003)
8. Sexton, T.: Dream Analysis of Carl Jung and Sigmund Freud: The Difference Lies in the Unconscious. http://voices.yahoo.com/dream-analysis-carl-jung-sigmund-freud-the-466946.html (2007). Accessed 21 March 2012, from Yahoo Voices