Using Natural Language Processing to Improve Document Categorization with Associative Networks

Niels Bloom

Pagelink Interactives, Sherwood Rangers 29, 7551 KW Hengelo, The Netherlands
University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands
n.bloom@pagelink.nl

Abstract. Associative networks are a connectionist language model with the ability to handle large sets of documents. In this research we investigated the use of natural language processing techniques (part-of-speech tagging and parsing) in combination with Associative Networks for document categorization and compare the results to a TF-IDF baseline. By filtering out unwanted observations and preselecting relevant data based on sentence structure, natural language processing can pre-filter information before it enters the associative network, thus improving results.

Keywords: Associative Networks, WordNet, Stanford Natural Language Parser, Natural Language Processing, Document Categorization

1 Introduction

Ordering large sets of documents, such as the articles in Wikipedia, into a hierarchical structure of categories allows users to find information more easily. However, the changing nature of such data offers special challenges in creating and maintaining that hierarchy.

An associative network is a connectionist language model that is capable of automatically categorizing libraries while term frequency-inverse document frequency or TF-IDF [6] is a method to determine how important a word is to a document in a set of documents. We wish to prove that associative networks outperform a TF-IDF baseline in document categorization and that the results can be further improved by using natural language processing or NLP-techniques for disambiguation and for detection of relevant text elements.

We tested associative networks without NLP, with a part-of-speech tagger and with full language parsing and compared the results to each other and to TF-IDF baselines. For the NLP we used the Stanford Natural Language Parser [3, 10]. In all cases, Princeton WordNet [2, 5] was used to link words to concepts.

2 Associative Network

An associative network is a mathematical model of associative thinking used by humans to solve certain problems [4, 7, 8]. It contrasts with case-based reasoning

approaches based on this work by not relying on formally defined cases and ontologies, but using a connectionist model as a base instead.

Associative networks are modelled as a graph: each node represents an observation, such as the occurrence of a specific concept, and each edge a link between these observations. Edges with higher weights represent more closely related observations. For example, the concepts of *fly* and *mosquito* are more closely related than *fly* and *insect*. Using the same connectionist foundation [1] as neural networks, each node can be activated, which spreads the input signal to the connected nodes proportionally to the edge weight. Thus, high weight edges spread the signal more strongly than low weight edges.

In our case, a document activates each word concept found in the document. Concepts that have more relevance to the document are activated more than others, thus spreading further and stronger through the associative network.

2.1 Creation of an Associative Network

To construct an associative network, we used Princeton WordNet [2, 5] as a foundation as it provides both a thesaurus and dictionary data in one system, is well documented and supported and is a well-accepted standard. The network was initialised by creating a graph consisting of a single node for each synset (word concept) in WordNet. These nodes where then connected according to the relationships between those synsets in WordNet. For example, the synset for *fly* has the sister-term *mosquito* and a hypernym *insect* so the nodes for these synsets were connected. All edge weights were initialised at a value of 1.

2.2 Training of an Associative Network

Not all relationships in WordNet are equal in conceptual distance and thus they should not have the same weight in an associative network. However, the number of steps to go from one synset to another does not directly determine the conceptual distance between the two. To make associative networks learn the actual conceptual distance between synsets, they can be trained using back-propagation to give more weight to closer connections and lower weight to more distant ones. As a result, when a node in that associative network is activated, activation is spread towards more closely related synsets more easily.

In our case, training was done by creating a training library composed of 30 manually selected Wikipedia articles with each article being closely related by topic to exactly one other article and not related to the other 28 articles, thus forming fifteen pairs of articles. Articles were selected from documents in the same category as the test set (see Section 3.1), but from different sub-categories.

A newly initialised associative network was activated for each of the 30 articles in random order to determine which were the most closely related. Depending on the correctness of the result, positive or negative reinforcement was applied by using back-propagation to adjust the weights in the network. This cycle was repeated until the associative network produced the correct matching article for the entire library. By training the associative network in this way, we

increased the weight of edges between closer concepts such as *fly* and *mosquito* while reducing the weight of edges to more distant concepts such as *insect*.

2.3 Improving the activation pattern

The quality of the activation pattern, that is the nodes in an associative network that are activated for a certain input, is important for getting good results. Our assumption is that we can use NLP-techniques to improve the quality of the input for associative networks.

To test the effect of NLP techniques, we used the Stanford Part-Of-Speech Tagger [10] to help exclude synsets which do not match the part of speech from the activation pattern and the Stanford Natural Language Parser [3] to boost the activation of key words in each sentence. Simply put, these techniques modify which nodes in the associative network are activated and by how much.

3 Design of the experiment

To prove the above assumption, we tested associative networks with three types of natural language processing: 1) a basic associative network without NLP, 2) an associative network with part-of-speech tagging, and 3) an associative network with a natural language parser.

The task was to sort 50 related articles into five predefined categories. The systems were evaluated by determining how well they matched the manual categorization made by the Wikipedia user base. Each system also calculated a baseline value using TF-IDF. In each of the cases, the natural language processing between the associative network and its TF-IDF baseline was identical.

3.1 Constructing the test sets

We created five test sets of 50 articles, each within a single category. The five categories were manually selected from Wikipedia, based on the criteria that they should be general topics with many articles and sub-categories. Categories selected included topics such as "Biology", "Philosophy" and "Nature".

Next, five sub-categories were randomly selected within each main category. Each sub-category had to contain at least ten qualifying articles and had to have a listed main article. An article qualified if it had a content of at least one thousand words, and if it was not marked as a stub or list article.

Finally ten qualifying articles were randomly selected from each of the five sub-categories, to give a test set of 50 articles. For example, the category "Biology" might have "Genetics", "Biochemistry", "Mycology", "Neuroscience" and "Ecology" as sub-categories. These were then mixed up randomly. For each article we wished to identify to which sub-category it belongs. To determine this, each sub-category's main article was used as a target, with which articles were compared to see how closely they matched.

3.2 Setup of the experiment

First, the text of each Wikipedia article in each test set was converted to three sets of synsets as described below. In each case, every synset in a set was given an activation value: the more important and frequent a synset was in relationship to the text, the higher the activation value. Once each set of synsets was constructed they were given to a trained associative network (see Section 2.2) as input, spreading the activation across the network.

The five target articles, each the main article of a sub-category, were connected in turn to the associative network, and received activation spread as the sets of synsets for each of the 50 articles were activated. The target article receiving the highest activation value was selected and the associated sub-category was determined as the one in which the test article should be categorized.

As mentioned, three methods were used to construct the set of synsets. Each system was made to determine which article matched which sub-category, based only on their textual content. No other information, such as links, was used. An accuracy score was established based on how many articles were sorted correctly. For example, if 40 articles were sorted correctly, the accuracy score would be 80%.

3.3 Methods to construct the set of synsets

Basic associative network. A basic associative network was established using WordNet [2, 5]. To determine a sub-category for each article, first the text in each article was split into words, which were matched with the corresponding synsets. If a word-form matched multiple synsets, it was not disambiguated, but all matching synsets were used. Thus, for each article a set of synsets was established, with a count of how many times each synset was matched.

These values were used as weights to connect the article to the associative network. Once all 50 test and five target articles were connected, each of the test articles was activated individually. The spread of this activation through the network activated each of the main articles to a different degree, depending on the available paths and the associated weights between the articles being activated. The amount of activation that spread to the main articles was then established, with the category of the one receiving the highest activation being selected as the category to place the test article in.

Associative network with part-of-speech tagger. Linking words to all synsets with the word as a word-form is not very precise. For example, the word *fly* in the sentence *I like to fly* matches both the act of *flying* and the insect *fly*.

To help correct for this, the Stanford Part-Of-Speech Tagger [10] was used to get a better match between words and synsets: using the default tagger, the type of each word was established. For every word, the synsets were filtered based on this type, with non-applicable synsets being removed. In the earlier example *fly* would be tagged as a verb, so it would not match the synset for the insect. The improved set of synsets was then used in the same way as described above to determine a sub-category for each article.

Associative network with natural language parser. Not every part of a sentence is equally important. For example, in the sentence *I want to fly to Paris in a jet*, the word *fly* is more important than the word *in*. We would like to know which words are more relevant before activating the network. Though the associative network can establish which concepts are important to some extent, if words that have less relevance in the sentence are activated less, the associative network will be better able to identify relevant concepts in turn.

Using the Stanford Natural Language Parser [3], each sentence in the documents was tagged and parsed to determine the dependency relations between its words. In the earlier example sentence seven connections are made, five of which include the word *fly*. This implies that the word *fly* is crucial in the sentence.

Based on this idea that words connected to many other words in the sentence are more relevant to the sentence, the weight of each word's associated synsets was increased with a factor of the total amount of connections in the sentence to the amount of connections involving that word. Once weighted, the same techniques as above were used to sort the articles into the different sub-categories.

3.4 TF-IDF baseline

Because TF-IDF is an unsupervised learning method, while associative networks are a supervised method, we expect the baseline to have a lower level of performance than associative networks. To establish the baseline, we calculated the TF-IDF value based on synsets (rather than raw words), as determined by the three methods described above. The resulting values were used to determine the distance between each test article and the five target articles. Each test article was assigned the sub-category of the target article to which it had the shortest distance. The distance was calculated using the following method:

$$D(a_x, m_y) = \sum_{n=1}^{s} |tfidf(a_x, S_n) - tfidf(m_y, S_n)| \quad (1)$$

where D(a_x, m_y) is the distance between the article a_x and the main article m of sub-category y. S is the set of synsets 1...s in the corpus and *tfidf(a, S_j)* is the term frequency / inverse document frequency of the synset S_j in article a.

4 Experimental results

	TF-IDF accuracy	AN accuracy
No NLP	34 %	78 %
Part-of-speech tagger	31 %	84 %
Natural language parser	34 %	87 %

Table 1. Results of the experiment

Table 1 lists the average results over five sets of 50 documents. The associative network approach clearly outperformed the TF-IDF baselines. Not only did the TF-IDF approach gain nothing from NLP, the results actually worsened, most likely because relevant words now activate fewer synsets: in turn fewer synsets match with the target articles. By contrast, associative networks clearly benefit from the improved match between word-forms and synsets, gaining six percent-point with part-of-speech tagging and another three using the parser.

5 Conclusions and Future Work

Because of the associations made by the associative network, terms relevant to the text are brought forward even if they are not mentioned in the text or are only mentioned a few times. As expected, the associative network connects better with the underlying meaning of the text than a TF-IDF based approach.

In this work, we have compared associative networks against a TF-IDF baseline. As future work, we plan to set them up against supervised learning methods [9] to see whether associative networks can outperform these as well.

NLP-techniques help to extract the meaning from the text, filtering out synsets that share a common word-form with the synset which actually corresponds to the word, and identifying key words in each sentence. As a result, the accuracy of the network in categorising articles is increased.

We have used only basic NLP-techniques in our tests, yet these have already increased the success of the associative networks. Further research into the combination of associative networks and NLP may improve the results even further.

References

1. Bechtel, W.: *Connectionism and the philosophy of mind: an overview.* The Southern Journal of Philosophy, 26: 1741, 1988
2. Fellbaum, C.: *WordNet: An Electronic Lexical Database.* Cambridge, MA: MIT Press, 1998
3. Klein, D., Manning, C.: *Fast Exact Inference with a Factored Model for Natural Language Parsing.* Adv. in Neural Information Processing Systems, 15: 3-10, 2003
4. Marcus, G. F.: *The Algebraic Mind: Integrating Connectionism and Cognitive Science.* Cambridge, MA: MIT Press., 2001
5. Miller, G.: *WordNet: A Lexical Database for English.* Communications of the ACM, 38 No. 11: 39-41, 1995
6. Ramos, J.: *Using TF-IDF to Determine Word Relevance in Document Queries.* Proceedings of the First Instructional Conference on Machine Learning (iCML), 2003
7. Schank R. C.: *Dynamic Memory: A Theory of Learning in Computers and People.* New York: Cambridge University Press, 1982
8. Schank, R.C., Abelson, R.P.: *Scripts, Plans, Goals and Understanding.* Erlbaum, Hillsdale, New Jersey, U.S., 1977
9. Sun, J., Chen, Z., Zeng, H., Lu, Y., Shi, C., Ma, W.: *Supervised latent semantic indexing for document categorization* Proceedings for ICDM: 535-538, 2004
10. Toutanova, K., Klein, D., Manning, C., Singer, Y.: *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network.* Proceedings of HLT-NAACL: 252-259, 2003