Semantic Based Text Classification Using WordNets: Indian Language Perspective

S. Mohanty and **P. K. Santi** and **Ranjeeta Mishra** and **R.N. Mohapatra** and **Sabyasachi Swain**
RC-ILTS- ORIYA
Post Graduate Department. of Comp. Sc. and Application
Utkal University,
Bhubaneswar, Orissa, India- 751004.
{sangham1, pksanti, mishra.ranjeeta, Sabyasachi_swain}@rediffmail.com

Abstract

Automatic text classification is an area that has received a great deal of attention in recent research due to current growth of Internet, which has resulted in huge amount of information that has become a challenge to access efficiently. This paper describes an experimental result on how to create an automatic efficient and effective tool that is able to classify large documents quickly. Our method is built on lexical chain of linking significant words that are about a particular topic with the help of hypernym relation in WordNet. We have tested for the Indian language Sanskrit using SanskritNet and also extracting and scoring lexical chain considering with necessary design decisions.

Introduction

Automatic text classification has received a great deal of attention in recent research due to the rapid growth of the Internet to access information effectively. This is to develop a computationally efficient system to create the required text automatically. Classification of document has been viewed as a two-step process. The first step is the extraction of important concepts from the source text by building an intermediate representation of some sort. The second step uses this intermediate representation to generate a summary. In the research presented here, we concentrate on the 1st step of the classification process and follow Barzilay and Elhadad (1997) in employing lexical chains to extract important concepts from a document. We present a linear-time algorithm for lexical chain computation and offer an evaluation, which indicates that such chains are a promising avenue of study as an intermediate representation in the classification process.

In this paper we present a computationally efficient method based on lexical chain using OriNet and SanskritNet taxonomy (S. Mohanty, 2002). The resulting lexical chains are a means of identifying cohesive regions in a text, with applications in many natural language processing tasks, including text classification.

1 Algorithm for Building Lexical Chain

To extract lexical chains from a source document using the semantic network of WordNet, we have done an interpretation as a mapping of noun instances to specific senses, and further, of these senses to specific lexical chains. Each unique mapping is a particular "way of interpreting" the document, and the collection of all possible mapping senses all of the interpretations possible. In order to compute lexical chains in linear time, instead of computing every interpretation of a source document as Barzilay and Elhadad did, a structure that implicitly stores every interpretation without actually creating it, thus keeping both the space and time usage of the program linear is created. Next the method for finding that interpretation which is best from within this representation is developed. As was the case with Barzilay and Elhadad, one has to rely on WordNet to provide sense possibilities for, and semantic relations among, word instances in the document. The lexical chain building process with the following steps.

1. Initialised set of Synset-ID as choice of domain name.
2. Tag the source document using POS Tagger and Morphological Analyser (S. Mohanty et al., 2004).
3. Select a set of candidate words with number of occurrence in the document. (Here we consider nominal word is the only candidate word).

Petr Sojka, Key-Sun Choi, Christiane Fellbaum, Piek Vossen (Eds.): GWC 2006, Proceedings, pp. 321–324. © Masaryk University, 2005

4. For each candidate word in the source document, generate synset-chain of each sense by looking up synonym and hypernym relations in the WordNet. This information is stored in an array indexed on the Synset-ID of the word from WordNet for constant time retrieval.

5. Generate the semantic tree and select the best synset-chain and the nodes (Synset-ID) of domain name which score will be affected most greatly by removing this node from it.

2 Experimental Result on the Sanskrit Document

Let us demonstrate the lexical chain procedure on the following paragraph in Sanskrit language. The italic words are nominal words known as candidate word retained to generate lexical chain.

2.1 Algorithm To Find the Appropriate Concept from a Text

1. The given text is stored in a text file.

2. Frequency of the nouns occurring in the text are identified and stored.

3. The lexical chain of each of these nouns are calculated with the help of WordNet and are stored as a list.

4. The leaf nodes with higher frequencies are considered in terms of their lexical chains.

5. Nodes with higher frequencies are matched through their lexical chains to find the concept.

6. The node with the capacity of accommodating the highest number

of leaf nodes provides the concept of the particular text.

In the given text, the higher frequency words i.e. पशु, पक्षि, मनुष्य and जीवनम् in sanskrit are identified and their corresponding Synset-IDs with their lexical chains are generated. We can find that all these noun words are meeting at a common node (304 for this text). That common node is considered to be the appropriate concept for the text. Here we can see that the concept generated for Sanskrit is चाक्षुषमुर्तद्रव्य gives us the same concept i.e Visual Mobile Substance.

3 Conclusion

We have outlined an efficient, linear-time algorithm for computing lexical chains of Sanskrit language as an intermediate representation for automatic machine text classification. The benefit of this algorithm is its ability to compute lexical chains in documents significantly larger than could be handled by Barzilay and Elhadad's implementation. Thus, our algorithm makes lexical chains a computationally feasible intermediate representation for classification.

References

Barzilay R. and Elhadad M. (1997), *"Using lexical chains for text summarization*", Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS-97), Madrid, Spain.

Mohanty S. and Santi P. K. (2002), *"Object Oriented Design Approach to OriNet System: Online Lexical Database for Oriya Language"*, IEEE Proceedings of LEC-2002, University of Hyderabad, Hyderabad, India.

Mohanty S., Santi P. K., Das Adhikary K.P. (2004), *"Analysis and Design of Oriya Morphological Analyser: Some Tests with OriNet"*, Proceedings of symposium on Indian Morphology, Phonology and Language Engineering, 2005, IIT Kharagpur, India.

The Sanskrit Text

मृत्तिकाः जलं पवनश्च पृथिव्याः साधनानि । अत्र **मनुष्य-पशु-पक्षिणः** निवसन्ति । **मनुष्यः गृहे** तिष्ठति । **पशु-पक्षिणः वने** तिष्ठन्ति। प्रत्येकस्मिन् **जीवशरीरे जीवात्मा** अस्ति। जीवात्मनः अन्यं **नाम जीवनम्**। **वृक्ष-लतानामपि जीवनम्** अस्ति। एते **सजीवाः** इति कथ्यन्ते।

Table-1: Candidate word with synset-ID and frequency of Sanskrit nouns.

Candidate Word	Root Word	Synset-ID	Freq of word in text
मृत्तिका	मृत्तिका	233,234	1
पवनश्च	पवनः	249	1
पृथिव्याः	पृथिवी	236	1
पशु	पशुः	258,260,263	2
पक्षिणः	पक्षि	264,266	2
साधनानि	साधनम्	250,251,253	1
मनुष्य	मनुष्य	255	2
गृहे	गृहं	268,272	1
जीव	जीव	259,277,279	1
शरीरे	शरीरं	251,281,282	1
जीवैत्मा	जीवात्मा	279	1
वृक्ष	वृक्षः	289	1
लतानामपि	लता	291,292	1
जीवनम्	जीवनम्	232,284,286	2
सजीवाः	सजीवः	259	1
वने	वनं	232,274,276	1
जलं	जलं	231,232,245	1

Table-4: Lexical Chain of the words of Sanskrit nouns.

Word	Synset-ID	Lexical Chain
मृत्तिका	233	233→235→236→237→238→239→240→241→254→242→304→303→302→300→243→100
	234	234→235→236→237→238→239→240→241→254→242→304→303→302→300→243→100
जलं	231	231→244→237→238→239→240→241→242→304→303→302→300→243→100
	232	232→238→239→240→241→242→304→303→302→300→243→100
	245	245→246→247→248→100
पवन	249	249→238→239→240→241→254→242→304→303→302→300→243→100
पशु	260	260→261→262→301→243→100
	263	263→256→257→258→259→304→303→302→300→243→100

पक्षि	264	264→265→257→258→259→304→303→302→ 300→243→100
	266	266→267→253→254→242→ 304→303→302→300→243→100
साधनम्	251	251→252→238→239→240→241→242→ 304→303→302→300→243→100
मनुष्य	255	255→256→257→258→259→ 304→303→302→300→243→100
गृहं	268	268→254→242→ 304→303→302→300→243→100
	272	272→273→100
जीव	277	277→278→273→100
	279	279→280→262→301→243→100
शरीर	282	282→283→281→239→240→241→254→242→ 304→303→302→300→243→100
वृक्षः	289	289→290→259→ 304→303→302→300→243→100
लता	291	291→290→259→ 304→303→302→300→243→100
	292	292→293→250→239→240→241→254→242→ 304→303→302→300→243→100
जीवनम्	232	232→238→239→240→241→254→242→ 304→303→302→300→243→100
	284	284→285→238→239→240→241→254→242→ 304→303→302→300→243→100
	286	286→287→288→250→239→240→241→254→242→ 304→303→302→300→243→100
सजीवः	259	259→304→303→302→300→243→100
वनं	274	274→275→238→239→240→241→254→242→ 304→303→302→300→243→100
	275	275→238→239→240→241→254→242→ 304→303→302→300→243→100