Machine Mapping of Document Collections: the Leximancer System.

Andrew E. Smith

School of Computer Science and Electrical Engineering
The University of Queensland
Queensland, Australia. 4072.

andrew.smith@member.sage-au.org.au

Abstract

A system is presented for rapidly mapping a text document collection using a set of concept dimensions. This offers an easy way to index and re-index large sets of documents using a flexible, faceted classification system of concepts, rather than just keywords.

Thesaurus-like concepts of interest are specified in advance, along with a few seed words to describe each. These concepts can be selected automatically if desired. A novel machine learning algorithm finds an optimal set. of weighted terms from the document set for each concept. The resulting thesaurusnetwork is used to classify and index the. collection, down to a possible sentence resolution. Weights of concepts and relationships between concepts are measured. and mapped on a concept cluster map. The. concept space can be. explored from an overview, down to the. document level.

Keywords Document Management, Information Retrieval, Text Mining.

1 Introduction

This paper presents a system for generating a conceptual index with minimal supervision. In contrast to normal keyword indexing methods, this system uses abstracted thesaurus concepts, and can offer a more precise and flexible alternative to whole-document classification and filing. In other words, this approach lies between keyword indexing and document classification, with the advantages of both. The method is efficient and requires minimal supervision.

2 Strategy

This project had as its goal the development of a practical Text Mapping and Exploration system. The strategy chosen to achieve this was based on a Concept Space approach (Chen et al [1]), in which words are mapped to a much smaller set of concepts:

1. Text preparation: Standard techniques are employed, including name and term preservation, tokenisation, and the application of a stoplist (eg. Miller et al [2]). No attempt at phrase binding is made. Stemming is not performed since automatic thesaurus generation makes this redundant, and undesirable.

2. Creation of Thesaurus Concepts: These can be devised in collaboration with a domain expert to suit the current requirements of the users, or they can be chosen automatically using a novel algorithm for finding significant seed words to reflect the themes present in the data.

3. Learning the thesaurus: Use a machine learning algorithm to find the optimal thesaurus words from the text data.

4. Classification: Classify the text using this thesaurus, to a possible sentence resolution.

5. Creation of a faceted two-level classification system: Designate primary concepts as entities and secondary concepts as properties of entities: index the tagged text using the entities and properties.

6. Mapping: Cluster the entities according to weight and relationship, to create a Concept Cluster Map.

7. User interface: Generate a simple hypertext browser for exploring the classification system in depth.

3 Machine Thesaurus Construction

The key to achieving concept mapping of text lies in the generation of thesaurus-like mappings from groups of words to concepts. The goal here is to accurately predict the likelihood that a fragment of text represents a subset of the concepts. This is often referred to as tagging.

3.1 Techniques for Word Cluster Formation

Traditional document clustering relies on document nearness metrics that use Single Term Indexing (Salton [4] sect 9.3). This treats a document descriptor as a 'bag of words' and ignores local structure. The next logical step is to perform term clustering based on the co-occurrence statistics of words in the collection: to cluster and retrieve documents (Salton [4] sect 9.4), and to construct a thesaurus (Salton [4] sect 9.6).

These Information Retrieval methods focus on document similarity, whereas this project is concerned with conceptual dimensions. This is a key discrimination, for the reason that most term clustering algorithms *induce* pattern features from the text which do not necessarily have clear conceptual meanings. These are useful for matching documents but do not. lead to a conceptual abstraction that can be tailored easily to the interests of the user. A principal goal of the work presented here was to allow the user or the system to tightly define each concept prior to learning and indexing the text body.

3.2 Techniques for Word Sense Disambiguation

As a result of the considerations mentioned in the previous section, the field of word sense disambiguation was examined for suitable methods. This discipline, part of Computational Linguistics, generally seeks to tag words with conceptual category indicators, such as thesaurus classes.

Many researchers have used static thesauri and lexicons to perform sense tagging. One of the most popular thesauri for this purpose is Wordnet (Miller et. al [2]). These methods suffer from the problem that the document vocabulary varies from time to time and from domain to domain. It is expensive to manually update or construct a thesaurus.

A second strand of research uses Bayesian Classifiers (Yarowsky [9]). or more generally. Linear Classifiers (Roth [3]). A Bayesian Classifier is a network of weighted mappings of observed features to likely scenarios. Yarowsky [9] presented a self-ordering algorithm for learning a network to disambiguate the different senses of specific words, such as 'plant'.

We developed a generalisation of Yarowsky's method using the following assertions:

- Words in text fragments constrain the choice of nearby words through the medium of a concept. This idea is borrowed directly from Corpus Linguistics (Stubbs [7]).
- As a result, we can generalise Yarowsky's method from senses of specific words to concepts in general.
- To achieve this, seed a concept with a small set of indicative words, or word combinations, taken from the training data. These words shoud be chosen wisely; the majority should only have meanings contained within the concept class, in the texts under inspection. In practice, a seed set of one word will work fine.

The details of the method are published in Smith [5].

4 Classification

Once the thesaurus network has been learnt, the classification procedure is simple:

1. Process the text sequentially in blocks of n sentences.
2. Look up all the words from the text block in the thesaurus network and add their weightings in each concept.
3. Threshold the results to select the relevant concepts with likelihood weightings.

This is very similar to coding, or sense tagging, as performed in Content Analysis (Weber [8]).

5 User Interface

A prototype system for indexing, mapping, and exploring these classified sentences was developed. The user interface was written in HTML. JavaScript, Java, and CSS. The 'Leximancer' system was designed for browsing the concepts and names present in a body of text. This system uses a two level classification system, where each level is faceted. At the top level are the concepts or names of primary interest. These are called entities. In Information Science terms they are absolute semantic dimensions. Relationship strengths are found in a pair-wise manner between each entity and all the others, which allows the number of entities to be reasonably large while avoiding serious combinatorial explosion.

The second level of the classification system consists of concepts or names which are called properties. These are also absolute dimensions, in that a property can apply to any entity. The properties associated with each entity are treated as facets. Only N-tuple combinations of properties which appear *in the context of each entity* are recorded, again to control combinatorial behaviour. It is emphasised that any concept can be designated as an entity, or a property, or both, depending on the information need of the user.

One of the principal aims is to quantify relationships between concepts. These relationships are then used to allow analysts to explore the text; that is, to approach a document collection like an unfamiliar continent. An overall ma]) of the concept space is created by clustering entities according to the strengths of their relationships. The algorithm used to produce the cluster map allows the concept points to order themselves on the surface of a sphere according to the strengths of the relationships between each pair (an example of the many body problem). The strength of each concept is indicated by its brightness. A screen shot is shown in Figure 1.

By clicking on the entity of interest, or by selecting from the pull down list, the user opens a hypertext browsing panel for that entity in the right half of the window. The entity panel lists the related entities, and the strengths of these relationships. This is a common approach in traditional data mining.

In addition, the associated properties of the concept can be viewed in this panel. In each of these panels which lists the intersections of two entities, or the intersections between an entity and multiple properties, the textual evidence can be browsed by clicking on the button image. Also, the vocabulary used in the text in association with this entity, and in association with each property of the entity, can be viewed.

6 Case Studies

A variety of real-world data sets have been analysed. These include sets of newspaper articles, a 50 Mb sample of Usenet news postings, a 100 Mb collection of job tracking list text data, the novel •Pride and Prejudice', the King James Bible, and 50 Mb of Federal Court judgements. At the time of writing, most of these case studies are viewable on the web [6]. These experiences have shown that this procedure is successful for learning and classifying from the same body of text. Expert analysts report high recall and precision from their case studies, with simultaneous precision and recall levels of over 90% being recorded. For the larger-data sets, learning was successfully performed on a sample of the data prior to classifying the whole. So far, no suitable benchmarking training data has been found. The Reuters-21578 collection is unsatisfactory because the tags are not embedded near their contextual words.

The algorithms perform efficiently, so mapping and remapping is not difficult. The most demanding phases are learning, indexing, and clustering.

The temporal asymptotic behaviour of the learning algorithm goes as the product of the length of the text body and the number of concepts. Memory usage is roughly proportional to the product of the size of the vocabulary and the number of concepts. The number of iterations for convergence of this algorithm does not depend on the number of words or concepts.

The indexing algorithm is memory limited. No simple proportionality can be assigned to this process, as memory usage depends on the number and nature of the concepts, the division of concepts into entities and properties, and also the textual indexing resolution and threshold chosen. The resolution and threshold can usually be adjusted to suit the available memory.

The temporal asymptotic behaviour of the cluster mapping algorithm is proportional to the square of the number of entity concepts. Memory usage is negligible.

For example, learning 40 concepts from 50 Mb of text, with a vocabulary of 18,000 words and 90,000 names, takes one hour without sampling. This data set has a disproportionately large vocabulary of names, which increases the memory footprint dramatically:

Hardware: Pentium III 500 Mhz single processor with 192 Mb RAM.

Operating System: RedHat Linux 6.1

Memory Footprint: 140 Ml)

Dedicated Runtime: 65 minutes

Cluster mapping 100 concepts takes 30 minutes. This is probably the largest number of points that should bo presented on one chart for cotnprehensibility:

Hardware: Pentium III 500 Mhz single processor with 192 Mb RAM.

Operating System: RedHat Linux 6.1

Memory **Footprint:** less than 10 Mb

Dedicated Runtime: 30 minutes

The final learning state is saved and this can be used next time a similar document set is analysed using the same concepts. This makes convergence much more rapid. Also, additional concepts can be learned at a later time and added to the thesaurus network.

7 Conclusion

It is proposed that, in some applications, such a process of conceptual mapping of document sets could be a better solution than filing or indexing. Conceptual mapping can offer less uncertainty to the user than keyword indexing, thus achieving better recall and precision. In contrast to whole document classification, mapping does not have to assign the same concepts to an entire document. More importantly, the technique presented here is relatively fast and largely unsupervised.

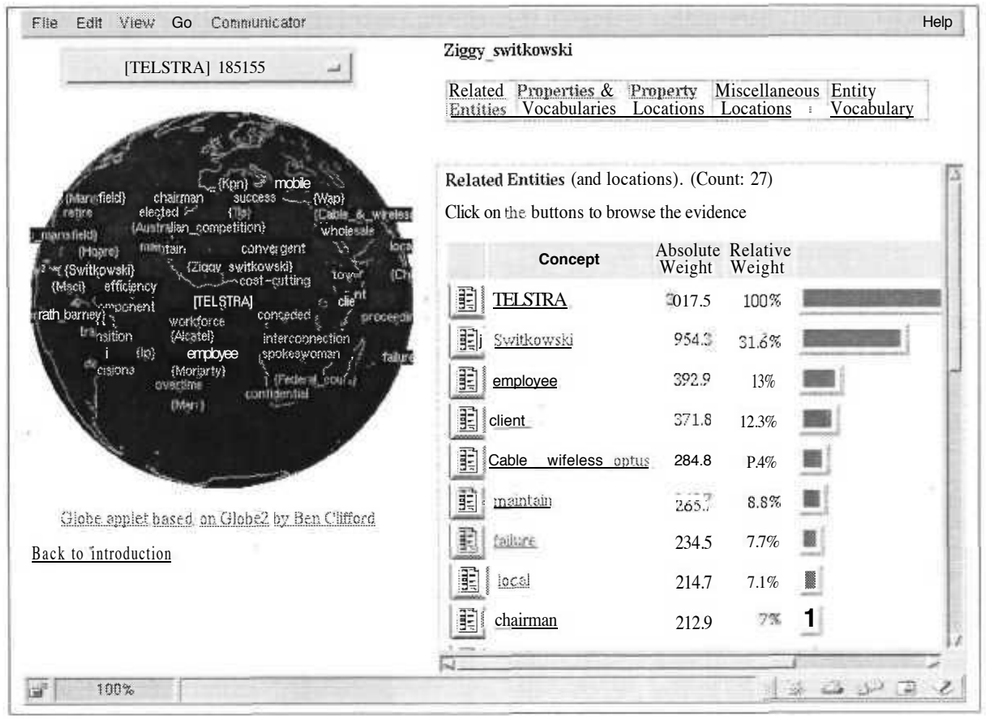

Figure 1: A screen shot of Leximancer.

References

[1] H. Chen. B.R. Schatz. T.D. Ng. J.P. Martinez, A.J. Kirchhoff and C. Lin. A parallel computing approach to creating engineering concept spaces for semantic retrieval: The illinois digital library initiative project. *IEEE Transactions on Pattern Analysis and Machine Intelligence. Special Section on ••Digital Libraries: Representation and Retrieval",* Volume 18, Number 8. pages 771 782. August 199G. http://ai.bpa.arizona.edu/go/report/technicalReport.html.

[2] George A. Miller, Richard Beckwith. Christiane Fellbaum, Derek Gross and Katharine .1. Miller. Introduction to wordnet: an on-line lexical database. *International Journal of Lexicography.* Volume 3. Number 4, pages 235-244. 1990. http://www.cogsci.princeton.edu/~wn/.

[3] Dan Roth. Learning to resolve natural language ambiguities: A unified approach. In *Proceedings of the Fifteenth National. Conference on Artificial Intelligence (AAAI-98),* pages 806-813, Madison, Wisconsin, 1998. http://12r.cs.uiuc.edu/-danr/publications.html.

[4] Gerard Salton. *Automatic Text Processing.* Addison-Wesley, 1989.

[5] Andrew E. Smith. Machine learning of well-defined thesaurus concepts. In *Proceedings of the International Workshop on Text and Web Mining (PRICAI 2000),* pages 72-79. Melbourne. Australia. August 2000.

[G] Andrew E. Smith. Text mining, automatic classification and indexing. http://www.csee.uq.edu.au/~aes/, July 2000.

[7] Michael Stubbs. *Text and corpus analysis : computer-assistedstudies of language and culture.* Blackwell Publishers, 199G.

[8] Robert Weber. *Basic ContentAnalysis.* Sage Publications. 1990.

[9] D. Yarowsky. Unsupervised word-sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for ComputationalLinguistics (ACL-95),* pages 189–196. Cambridge. MA. 1995. http://www.cs.jhu.edu/~yarowsky/pubs.html.