JUMP-STARTING CONCEPT MAP CONSTRUCTION WITH KNOWLEDGE EXTRACTED FROM DOCUMENTS

Alejandro Valerio & David Leake, Computer Science Department, Indiana University, USA
email:{avalerio, leake}@cs.indiana.edu

Abstract. Online documents provide a rich information resource for aiding the generation of concept-map-based knowledge models, but analyzing resources to select concepts and links is a time consuming task. This paper describes ongoing research on harnessing the information in unstructured textual documents, using information extraction algorithms, to generate a preliminary version of a concept map from a text document, for human refinement. The paper presents strategies for this task, implemented in an initial algorithm. The concept extraction phase of the algorithm has been evaluated with encouraging results.

1   Introduction

Given the large amount of information available online and the effectiveness of modern Web search engines, documents on the Web provide a useful starting point for humans gathering information about particular domains. Concept maps provide a useful medium for sharing such information in a concise and accessible form, making knowledge models based on concept maps (Novak & Gowin, 1984) a promising vehicle for knowledge sharing. Unfortunately, generating such knowledge models may require considerable effort to determine which concepts and relations to include. In some settings, this effort carries its own benefits, as when students increase their own understanding by performing the knowledge modeling process. However, to further the deployment of knowledge models, especially for large domains, it is desirable to facilitate the concept map generation process, through the use of automated systems to support knowledge model generation (e.g., Cañas et al. 2003, 2004b, Leake et al. 2004).

Natural Language Processing techniques are successfully used to automatically extract information from unstructured text documents through a detailed analysis of their content, often to satisfy particular information needs, such as providing a list of facts related to some event or topic or a specific answer to a question (e.g., Jackson & Moulinier, 2002). In this paper, we present ongoing research on applying such methods to automatically help concept map builders to exploit on-line documents, by applying information extraction procedures to automatically produce a preliminary version of a concept map, as a starting point for humans to adapt and eventually integrate into a knowledge model. The current goal of this work is not to produce a human-quality concept map from a given document automatically, which is a hard problem for long-term research. Instead, the goal is to perform preprocessing to speed up the generation of maps and integration of document information, for later refinement by a person or by another automatic process. At the same time, we expect that steps towards the current goal will contribute towards the long-term goal of autonomous concept map generation.

The paper is organized as follows: Section 2 presents some characteristics of concept maps which must be reflected in the document processing algorithm. Section 3 presents our algorithm and summarizes its current implementation. Section 4 describes an evaluation of the system, focusing on the concept extraction process which we see as the core of the required process. Section 5 presents some related work and Section 6 discusses some future directions for this project.

2   The Nature of Concept Maps

Ideally, automatically-generated concept maps would resemble the concept maps which might be constructed by a person with access to the same information. Consequently, we first consider already-known general features of "well-constructed" human-made concept maps, to guide design and evaluation. Novak and Cañas (2006) identify a number of concept map characteristics, involving both structure and content of the map, each of which has ramifications for automatic concept map generation.

Hierarchy of concepts: Besides extracting the concepts, a generation algorithm must sort them relative to their importance to the source document.

Different levels of abstraction and detail: Concept maps may be written in different levels of abstraction and detail depending on many conditions, i.e. expertise of the user or desired level of complexity. An ordered concept list may also be used to select the desired level of detail.

The information in propositions: In its simplest form, a concept map is a set of propositions. The concepts are mostly noun phrases, ideally consisting of few words that represent a particular concept. Concepts usually represent physical or abstract objects or entities; as a result, their labels usually contain nouns and adjectives. The linking phrases are usually verb or prepositional phrases that link concepts, so they usually have verbs and adverbs. A proposition is not necessarily a full sentence. This indicates that high granularity may be required when parsing the text.

Subjectivity: Both a concept map and a document represent the understanding or perspective of the author on a particular subject; the goal of generation is to reflect that perspective.

Informality: Similarly to documents, concept maps are not intended to provide formal presentations of concepts, definitions and semantic relations with other concepts. Therefore, the knowledge provided by the author may be captured in an informal textual form, without extracting formal relationships.

3 Extracting Knowledge from a Document to Build a Concept Map

Natural language processing techniques are used successfully on a variety of information extraction tasks (Harabagiu et al., 2005; Orkut et al., 2001; Alves et al., 2002). Each of these tasks requires a different treatment of the source depending on the particular information need and, in some cases, previous knowledge about the target context. Part of our research examines how to adapt and use the existing information extraction algorithms to our concept mapping task. For example, developing a useful concept map may require deciding how to select the granularity of entities (e.g., *"President of The United States of America"* might be best represented as a single concept in some contexts, and in terms of concepts for *"President," "United States,"* and *"America"* in others). An interesting long-term research question is how the requirements for language processing to jump-start concept map generation differ from those of more traditional NLP tasks. For example, when the goal is interactive support, perfect analysis is not needed, but, the suggestions must be sufficiently useful for the knowledge modeler to accept and benefit from the system.

Analyzing a document to extract information for a concept map requires two main extraction steps. The first is to identify and name the concepts in the text. Here we refer to "concepts" in a broad sense, meaning all the objects and entities mentioned in the document. These appear primarily as objects and direct/indirect objects in sentences, and will be noun phrases. The second is to identify and name relationships for linking phrases. We expect these to be suggested by possible syntactic dependencies between objects, with the connecting verb phrases in the sentences containing possible semantic relations which may correspond to the linking phrases of a future concept map.

The two extraction processes need not be solely bottom-up; they could use existing information about the target knowledge model or incoming document stream to bias the acquisition of concepts and their relations. However, in this paper, we focus on extracting information based on the document alone. This facilitates evaluation by eliminating potential sources of bias and noise based on background knowledge, and tests the domain-independent capabilities of the extraction algorithm to provide the first iteration of a concept map that could later be expanded and refined by a person or by automatic methods.

As a starting point, our current work focuses on the problem of generating a single concept map from a single document. This is a simplification because large documents may contain different "topics" or "themes" appropriate for producing several concept maps, and, several documents may refer to the same topic, making them reasonable to merge into a single concept map. We expect that an algorithm for the *1-1*

case will be scalable to such *n*-to-*m* transformations, by introducing a topic segmentation step that may be included before the process starts, splitting the document into *n* documents. Likewise, merging of information across maps could be addressed by having common concepts and relations merged into *m* distinct concept maps once all individual documents are processed. The problem of processing raw text documents is independent to the problem of splitting and merging concepts into coherent, independent concept maps.

We assume the incoming document is "well-written" and that it contains the description of a concept or set of concepts. Figure 1 summarizes the steps of our current algorithm. The steps of the algorithm, and the specific techniques we have applied in their implementation, are described in more detail below.
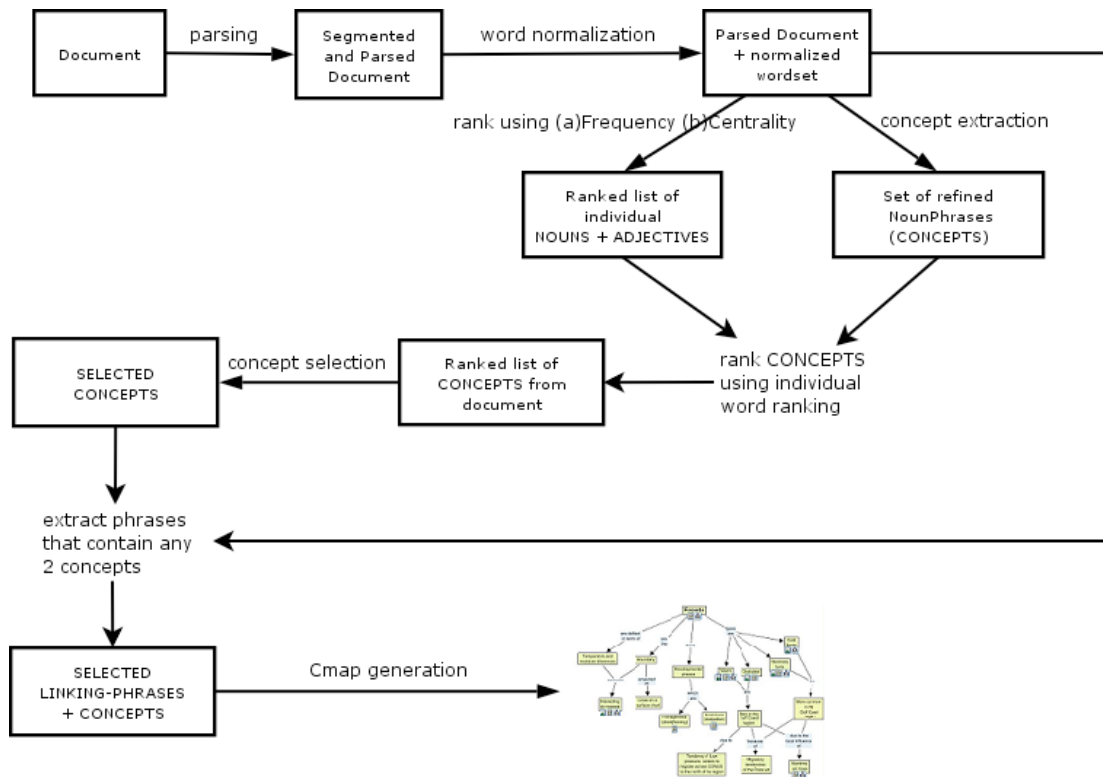

Figure 1. Procedure to construct a concept map automatically from a document.

Step 1. Document segmentation and parsing: The document is first preprocessed by an ad hoc sentence boundary detection algorithm based on regular expressions. Next, each individual sentence is parsed using Charniak's algorithm for a deep syntactic analysis (Charniak, 2000; Charniak et al., 2005). This step simultaneously tags each word with its part of speech, produces a parse tree for the sentence, and exposes the dependencies between the constituents. A shallow parser with linkage information could have been used in this step, but the subtree for each noun phrase is useful for adjusting the level of granularity when generating the concepts.

Step 2. Word normalization: Documents contain morphological variations of words that refer to the same entity, and may use multiple synonyms. The normalization step splits words into disjoint equivalence classes, in which two words *a* and *b* are considered equivalent if

(1) $POS(a) = POS(b)$

(2) $stem(a) = stem(b)$ or *a synonym b*,

where *POS(a)* is the part-of-speech of *a*: noun, verb, adjective,… and *stem(a)* is the basic form of the word after all affixes are removed. Porter's (1980) algorithm is used for word stemming and WordNet (Fellbaum, 1998) to determine the synonymy relation. We note that occasional conflicts may arise from the definition of equivalence, because the synonym relation is not transitive. Consequently, the definition may potentially

consider a word to belong to two different sets; in that case, the algorithm arbitrarily assigns it to either one. We do not expect this to have a significant impact in performance, because there is a 50% chance of picking the right set and this situation is not expected to occur frequently. Once the algorithm identifies the word equivalences, it tags each word with its class, for use to compare words in later steps.

Step 3. Individual word ranking: The next step, assigning weights to individual words, is needed for the later calculation of the weights of noun phrases. Several term weighting methods have been proposed for calculating term importance (Salton et al., 1988; Hulth, 2003), most relying on the relative importance of terms across a collection of documents. Such methods are not suitable for our task, because our aim is to develop methods to process the source document independently from other incoming documents or documents in the knowledge model, as a step towards solutions usable to jump-start the earliest steps of concept map generation, when additional information sources will not be available. Given this constraint, we used simple term frequency in the document, a method which has achieved some success when additional context information is unavailable (Salton et al., 1988). Our system considered only nouns and adjectives, as those are the most common parts of speech in concept-describing phrases. The analysis produces a list of words that is ordered by descending weight.

Step 4. Concept extraction: Starting from the parsed document and the normalized word set, the algorithm produces a set of noun phrases which will become the candidate concepts. The approach to this problem takes into account that: (1) ideally concept labels consist of a small set of words and (2) similarly to the word normalization step, noun phrases in a document with different surface forms may refer to the same concept.

Our algorithm starts by finding noun phrases that are closer to the leaves, without nested complex structures. In a parse tree, these are complex noun phrases, typically the object or direct object of a sentence, containing smaller noun phrases as its constituents. Although these large noun phrases reflect the dependencies of the sentence better, we need the smaller noun phrases instead because concept labels usually have a reduced number of words.
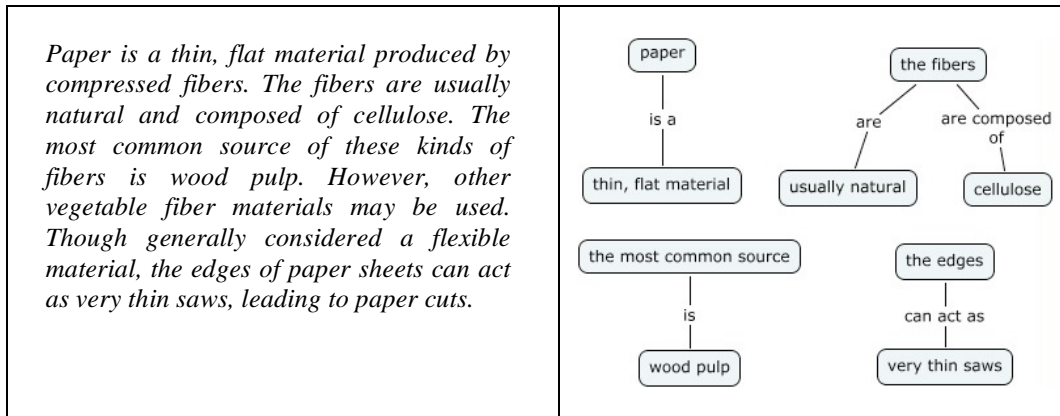
After the "minimal" noun phrases have been identified, the algorithm finds possible equivalences between them. It considers two noun phrases equivalent if (1) all nouns and adjectives in one are contained in the other and (2) this equivalence is not in conflict with any other. For example, given {*"President"*, *"President of Russia"*}, *"President"* and *"President of Russia"* are considered equivalent, but this is not true for {*"President"*, *"President of Russia"*, *"President of France"*} because the equivalence conflicts with *"President of France"*.

Step 5. Ranking concepts using the individual word ranking: Next, the set of extracted noun phrases is ranked using the ordered list of nouns and adjectives. This ranking ultimately will help selecting appropriate concepts to produce a clearer and more complete concept map. The rank of a noun phrase is given by the highest rank of its nouns and adjectives, with ties broken by the rankings of other words in the phrase if possible. The position of a concept in the ranking represents the relevance to the document's content.

Step 6. Concept selection: For short focused documents, it may be desirable to generate maps corresponding directly to the document itself. For longer or less focused documents, it may be desirable to limit the number of concepts presented in order to avoid overwhelming the user. A number of heuristics were assessed with informal tests: select the first n concepts, select a number of concepts incrementally until the concept map reaches certain size or cohesiveness, etc. The evaluation in the following section does not perform concept selection; instead, all concepts are used to construct the concept map.

Step 7. Linking phrase extraction: This phase tags the original parse trees with the selected concepts, extracting all pairs of concepts that have an indirect dependency link through a verb phrase. We presume that these phrases show relations between the concepts, so we extract those as the linking phrases. These phrases are marked in the trees.

Step 8. Concept map generation: The final step gathers the information from the tagged parse trees to construct the concept map: the selected noun phrases become the concepts and the connecting verb phrases are the linking phrases. The current system does not produce a graphical representation of the concept map (layout is a non-trivial issue). However, some auto-layout capabilities are already implemented on concept mapping tools, as in CmapTools (Cañas et al., 2004a) from the Institute for Human and Machine Cognition, which could be used to produce an initial graphical representation of the concept map.

Paper is a thin, flat material produced by compressed fibers. The fibers are usually natural and composed of cellulose. The most common source of these kinds of fibers is wood pulp. However, other vegetable fiber materials may be used. Though generally considered a flexible material, the edges of paper sheets can act as very thin saws, leading to paper cuts.

Figure 2. Example of system output from processing a short document

Figure 2 illustrates the algorithm's output. All concepts linked by the algorithm are shown; isolated concepts were removed for legibility. Note that some prepositional forms in the source document are not yet being used as linking phrases, which is a current deficiency of the algorithm and part of our future work.

4   Experimental setup and Results

Our evaluation focuses on Steps 1 through 5 in the previous section, testing the algorithm's performance extracting an ordered list of concepts as the starting point for constructing a concept map. To test the quality of the concept list produced, we started from an existing knowledge model already annotated with documents, and tested the ability of the algorithm to recover the concept map from the documents alone. Each document is processed individually, with its concepts automatically extracted using the described procedure. The evaluation is then based on match between the identified concepts and the set of concepts to which each document was originally attached. Although we are considering this as a portion of the general concept generation process, we note that the test task is important task in its own right: It could be used as to automatically index documents for a multimedia knowledge model, by taking a document and suggesting concept map nodes to which the document should be attached.

The experimental design is based on some premises about the association between concepts and attached documents in the expert knowledge models. First, we assume that if a document was attached to a given concept C in the expert's knowledge model, it is because the document contains either a description of C or relevant information about it, and therefore that C must be one of the concepts which should be ranked highly in the document. Second, we assume that each document is linked to the most related set of concepts in the enclosing concept map, but to no others. In other words, a concept C would not link to a document that is not relevant, and if some document is relevant to C, then a link between them will exist. These assumptions may not always hold, because we are processing data that may have errors and noise, but those are conditions that can generally be found in well constructed knowledge models.

In the test, a document D is selected for attachment to a concept C if C occurs in the first n concepts extracted from D. If the document is attached correctly (i.e., the attachment matches the choice made in the initial knowledge model), this supports the quality of the ranking in the list, because (1) the concept is in the list and (2) the concept appeared as one of the first concepts in order of relevance. We note that this

may be seen as a stringent test, because the algorithm is only credited if it replicates the assignment in the original knowledge model, even though it is possible that alternative placements would be acceptable.

As test data, we selected 80 documents at random from those in STORM-LK knowledge model (Hoffman, R. R. et al., 2001). The system processed each document to extract the concept ranking, and, the documents were "reattached" to the concepts in the knowledge model as described above. Because our procedure permits documents to be attached to multiple concepts, we used the average precision/recall for all the documents to evaluate the results. We evaluated the algorithm in two ways: (1) using only the concept that ranked first and (2) using the concepts that ranked first and second. The algorithm was tested with and without the stemming procedure to measure the impact of this normalization. Our algorithm's performance was compared to a baseline algorithm, which attaches a document to a concept if the concept contains any of the nouns or adjectives in the document.

Baseline		Without stemming – Top 1		With stemming – Top 1	
Precision	Recall	Precision	Recall	Precision	Recall
0.267	0.890	0.467	0.537	0.578	0.712

Baseline		Without stemming – Top 2		With stemming – Top 2	
Precision	Recall	Precision	Recall	Precision	Recall
0.267	0.890	0.408	0.707	0.553	0.848

Table 1. Results for algorithm evaluation

Table 1 shows the results of the testing. The results suggest that using noun phrases produced by the algorithm increases precision compared to using the individual words from the baseline. Stemming increased overall performance, and precision was improved by using only the top-1 or top-2 concepts. Even when the matching is so strict, the recall average is between 53% and 84%. In general, the algorithm successfully associated the documents to the original concepts, which is promising for its use as the basic domain independent algorithm to extract concepts.

We note that because the final goal of the project is to aid generation of new concept maps, rather than to attach documents to concept maps, the algorithm did not use information in the maps to provide top-down guidance. We expect that performance for the document annotation task would be improved by using such information.

5   Related work

5.1   Concept generation to provide suggestions to aid concept map generation

Recent research has aimed at retrieving relevant documents from the web (Leake et al., 2004) and at annotating existing knowledge models with those documents (Reichherzer & Leake, 2006). However, this work does not aim to extract individual concepts from documents, instead focusing only on a global match between document terms and concept labels. A number of successful concept suggesters have been developed to provide the user with a list of relevant terms to extend a concept map (Cañas et al., 2004b; Leake et al., 2004). Because a byproduct of our algorithm's processing is a list of concepts extracted from a document, ordered by relevance, our approach might potentially form the basis of another suggester of concepts, as discussed later in this paper; the key difference that our approach is not term-based but "concept-based".

5.2   Other solutions to produce concept maps from documents

Some prior work attempts to construct concept maps or similar representations automatically from text. Alves A. et al. (2002) uses WordNet to extract an initial hierarchy of nouns from a document to build an initial list of concepts, followed by several user feedback iterations to deduce relationships between pairs of concepts and hypothesize about their relations. Clariana, R. et al. (2004) present an approach which relies

on a predefined list of domain specific concepts provided by an expert. It considers two concepts to be related if they occur in the same sentence, but does not suggest possible linking phrases. On the other hand, Rajaraman K. et al. (2002), focuses word sense disambiguation. Once the meaning of nouns and verbs are resolved, it searches for Noun-Verb-Noun structures in the sentences, which become the concept – linking phrase relations.

Our approach is different in two important aspects. It uses the syntactic structure of the sentences and dependency information to find relations between the words. The relations are not retrieved from predefined ontologies, but are generated from the document itself. This enables the approach to be applied in any domain, without initial knowledge capture. In addition, it is more sensitive to the intentions of the document author. For example, even if two concepts are related in a particular the author might have intentionally ignored that relation, because it did not correspond to the desired level of abstraction. In addition, our algorithm produces concepts based on noun phrase structures rather than on individual words, making the concept labels more complete; capturing these from documents may also make them closer to the concept descriptions produced by a person. Such subjective characteristics, as well as the overall usefulness of the system to support users, are subjects for future human subject evaluations.

6   Summary and Future Work

This paper presents ongoing research on bootstrapping the concept map generation process, by generating preliminary concept maps based on documents. We have fully implemented an initial algorithm for this process, and have evaluated one central component, a domain-independent algorithm to extract from a document a list of concepts, ranked by their relevance to the source document. The algorithm uses a deep syntactic parse to select of noun phrases with different levels of granularity when desired. A basic term weighting approach is used in this first version of the process, and is a target for refinement, but an initial evaluation of the algorithm shows encouraging results.

In future work, we expect to study and refine aspects of the algorithm such as the term weighting approach for individual words. For example, we have done initial tests of an algorithm proposed by Mihalcea et al. (2004), which constructs a graph representation of the document to compute individual term scores using a modified version of PageRank. While results were mixed, we plan to explore this in further. We also plan to evaluate the impact of alternative normalization procedures on performance, such as name-entities co-reference and anaphora resolution. By the same token, a more accurate solution to conflicting synonyms might be to use a word sense disambiguation algorithm, but only for the words in conflict. There is no need to disambiguate the sense of all words, as we assume that if no synonym conflict exists the word was placed in the correct set. An algorithm has been developed to disambiguate words in concept maps (Cañas et al., 2003), but it cannot be used at this point because for our task, the conflicts need to be solved before the concept map is formed. In addition, we intend to include other syntactic forms as candidate linking phrases, to test the linking phrase extraction phase, which we expect to require a human subjects experiment, and to explore other heuristics to weight the noun phrases based on the term weights.

Once the algorithm is refined, we will consider two problems. The first is how to use the information extracted from a set of related documents to produce a set of concept maps. This may require the segmentation of documents in smaller topics or themes, and also the identification of common themes across documents. The second is how to apply additional information beyond the document itself. This information will come from (1) the concept maps in the target knowledge model, to bias the extraction of concepts and linking phrases and (2) crossed information on other related documents and documents existing in the knowledge model. We expect this information not only to allow more accurate construction of the concept map, but also to further the possibility of automatically attaching documents to concepts, if desired.

7   Acknowledgements

8 References

Alves, A., Pereira, F., Cardoso, A. (2002). *Automatic Reading and Learning from Text.* Proceedings of the International Symposium on Artificial Intelligence.

Cañas, A. J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., et al. (2004a). CmapTools: A Knowledge Modeling and Sharing Environment. In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping* (Vol. I, pp. 125-133). Pamplona, Spain: Universidad Pública de Navarra.

Cañas, A. J., Carvalho, M., Arguedas, M., Leake, D. B., Maguitman, A., & Reichherzer, T. (2004b). Mining the Web to Suggest Concepts during Concept Map Construction. In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology. Proc. of the 1st Int. Conference on Concept Mapping.* (Vol. I, pp. 135-142). Pamplona, Spain: Univ. Pública de Navarra.

Cañas, A. J., Valerio, A., Lalinde-Pulido, J., Carvalho, M., & Arguedas, M. (2003). *Using WordNet for Word Sense Disambiguation to Support Concept Map Construction.* Proceedings of SPIRE 2003: International Symposium on String Processing and Information Retrieval, Manaus, Brasil.

Charniak, E. (2000). *A Maximum-Entropy-Inspired Parser.* Proceedings of NAACL-2000.

Charniak, E. & Johnson, M. (2005). *Coarse-to-fine n-best parsing and Maximum Entropy discriminative reranking.* ACL'05.

Clariana, R. et al. (2004) *A Computer-Based Approach for Translating Text into Concept Map-like Representations.* In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping* (Vol. I, pp. 125-133). Pamplona, Spain: Universidad Pública de Navarra.

Fellbaum, C,. ed. (1998), *WordNet: An Electronic Lexical Database*, MIT Press.

Harabagiu S., Moldovan D., Clark C., Bowden M., Hickl A., Wang P. (2005). *Employing Two Question Answering Systems in TREC 2005*. Proceedings of the 14th Text Retrieval Conference.

Hoffman, R. R., Coffey, J. W., Ford, K. M., and Carnot, M.-J. (2001). *Storm-LK: A human-centered knowledge model for weather forecasting*. The 45th Annual Meeting of the Human Factors and Ergonomics Society, Minneapolis, MN, USA.

Hulth, A. (2003). *Improved automatic keyword extraction given more linguistic knowledge*. The 2003 Conference on Empirical Methods in Natural Language Processing, Japan.

Jackson, P. & Moulinier, I. (2002) *Natural Language Processing for Online Applications*. Philadelphia: John Benjamins Publishing.

Leake, D. B., Maguitman, A., Reichherzer, T., Cañas, A. J., Carvalho, M., Arguedas, M., et al. (2004). Googling from a Concept Map: Towards Automatic Concept-Map-Based Query Formation. In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping* (Vol. I, pp. 409-416). Pamplona, Spain: Universidad Pública de Navarra.

Mihalcea, R. & Tarau, P. (2004). *TextRank:Bringing order into texts*. In Proceedings of EMNLP.

Novak, J. D. & A. J. Cañas (2006). *The Theory Underlying Concept Maps and How to Construct Them.* Technical Report IHMC CmapTools 2006-01. Florida Institute for Human and Machine Cognition.

Novak, J. D., & Gowin, D. B. (1984). *Learning How to Learn*. New York: Cambridge University Press.

Orkut, B., Garcia-Molina, H., Paepcke, A. (2001). *Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices*. The 10th International WWW Conference. Hong Kong, China.

Porter, M.F. (1980). *An algorithm for suffix stripping*, Program, Vol 14, Number 3. pp 130−137.

Rajaraman, K. & Ah-Hwee Tan. (2002). *Knowledge Discovery from Texts: A Concept Frame Graph Approach*. The 11th International Conference on Information and Knowledge Management.

Reichherzer, T. & Leake, D. (2006). *Towards Automatic Support for Augmenting Concept Maps with Documents*. In A. J. Cañas, J. D. Novak (Eds.), Concept Maps: Theory, Methodology, Technology, Proc. of the 2nd Int. Conf. on Concept Mapping. San Jose, Costa Rica: Universidad de Costa Rica.

Salton, G. & Buckley, C. (1988). *Term-weighting Approaches in Automatic Retrieval*. Information Processing and Management. Vol 24, Number 5.