Redundancy: helping semantic disambiguation

Caroline Barrière
School of Information Technology and Engineering
University of Ottawa
Ottawa,Canada,K1N 7Z3
barriere@site.uottawa.ca

Abstract

Redundancy is a good thing, at least in a learning process. To be a good teacher you must say what you are going to say, say it, then say what you have just said. Well, three times is better than one. To acquire and learn knowledge from text for building a lexical knowledge base, we need to find a source of information that states facts, and repeats them a few times using slightly different sentence structures. A technique is needed for gathering information from that source and identify the redundant information. The extraction of the commonality is an active learning of the knowledge expressed. The proposed research is based on a clustering method developed by Barrière and Popowich (1996) which performs a gathering of related information about a particular topic. Individual pieces of information are represented via the Conceptual Graph (CG) formalism and the result of the clustering is a large CG embedding all individual graphs. In the present paper, we suggest that the identification of the redundant information within the resulting graph is very useful for disambiguation of the original information at the semantic level.

1 Introduction

The construction of a Lexical Knowledge Base (LKB), if performed automatically (or semi-automatically), attempts at extracting knowledge from text. The extraction can be viewed as a learning process. Simplicity, clarity and redundancy of the information given in the source text are key features for a successful acquisition of knowledge. We assume success is attained when a sentence from the source text expressed in natural language can be transformed into an unambiguous internal representation. Using a conceptual graph (CG) representation (Sowa, 1984) of sentences means that a successful acquisition of knowledge corresponds to transforming each sentence from the source text into a set of unambiguous concepts (correct word senses found) and unambiguous relations (correct semantic relations between concepts).

This paper will look at the idea of making good use of the redundancy found in a text to help the knowledge acquisition task. Things are not always understood when they are first encountered. A sentence expressing new knowledge might be ambiguous (at the level of the concepts it introduces and/or at the level of the semantic relations between those concepts). A search through previously acquired knowledge might help disambiguate the new sentence or it might not. A repetition of the exact same sentence would be of no help, but a slightly different format of expression might reveal necessary aspects for the comprehension. This is the avenue explored in this paper which will unfold as follows. Section 2 will present briefly a possible good source of knowledge and a gathering/clustering technique. Section 3 will present how the redundancy resulting from the clustering process can be used in solving some types of semantic ambiguity. Section 4 will emphasize the importance of semantic relations for the process of semantic disambiguation. Section 5 will conclude.

2 Source of information and clustering technique

To acquire and learn knowledge from text for building a lexical knowledge base, we need to find a source of information that states facts, and repeats them a few times using slightly different sentence structures. A technique is needed for gathering information from that source and identify the redundant information.

These two aspects are discussed hereafter: (1) the choice of a source of information and (2) the information gathering technique.

2.1 Choice of source of information

When we think of learning about words, we think of textbooks and dictionaries. Redundancy might be present but not always simplicity. Any text is written at a level which assumes some common knowledge among potential readers. In a textbook on science, the author will define the scientific terms but not the general English vocabulary. In an adult's dictionary, all words are defined, but a certain knowledge of the "world" (common sense, typical situations) is assumed as common adult knowledge, so the emphasis of the definitions might not be on simple cases but on more ambiguous or infrequent cases. To learn the basic vocabulary used in day to day life, a very simple children's first dictionary is a good place to start. In (Barrière, 1997), such a dictionary is used for an application of LKB construction in which no prior semantic knowledge was assumed. In the same research the author explains how to use a multistage process, to transform the sentences from the dictionary into conceptual graph representations. This dictionary, the **American Heritage First Dictionary**[1] **(AHFD)**, is an example of a good source of knowledge in terms of simplicity, clarity and redundancy. Some definitions introduce concepts that are mentioned again in other definitions.

2.2 Gathering of information

Barrière and Popowich (1996) presented the idea of concept clustering for knowledge integration. First, a Lexical Knowledge Base (LKB) is built automatically and contains all the nouns and verbs of the AHFD, each word having its definition represented using the CG formalism. Here is a brief summary of the clustering process from there. It is not a statistical clustering but more a "graph matching" type of clustering. A trigger word is chosen and the CG representation of its defining sentences make up the initial CCKG (Concept clustering knowledge graph). The trigger word can be any word,

but preferably it should be a semantically significant word. A word is semantically significant if it occurs less than a maximal number of times in the text, therefore excluding general words such as place, or person. The clustering is really an iterative forward and backward search within the LKB to find definitions of words that are somewhat "related" to the trigger word. A forward search looks at the definition of the words used in the trigger word's definition. A backward search looks at the definition of the words that use the trigger word to be defined. A word becomes part of the cluster if its CG representation shares a common subgraph of a minimal size with the CCKG. The process is then extended to perform forward and backward searches based on the words in the cluster and not only on the trigger word.

The cluster becomes a set of words related to the trigger word, and the CCKG presents the trigger word within a large context by showing all the links between all the words of the cluster. The CCKG is a merge of all individual CGs from the words in the cluster.

Table 1 shows examples of clusters found by using the clustering technique on the AHFD. If a word is followed by _#, it means the sense # of that word. The CCKGs corresponding to the clusters are not illustrated as it would require much space to show all the links between all the words in the clusters.[2]

The clustering method described is based on the principle that information is acquired from a machine readable dictionary (the AHFD), and therefore each word is associated with some knowledge pertaining to it. To extend this clustering technique to a knowledge base containing non-classified pieces of information, we would need to use some indexing scheme allowing access to all the sentences containing a particular

---

[2]The reader might wonder why such word as [rainbow] is associated with [needle_1] or why [kangaroo] is associated with [stomach]. The AHFD tells the child that *"A rainbow looks like a ribbon of many colors across the sky."* and *"Kangaroo mothers carry their babies in a pocket in front of their stomachs."* The threshold used to define the minimal size of the common subgraph necessary to include a new word in the cluster is established experimentally. Changing that threshold will change the size of the resulting cluster therefore affecting which words will be included. The clustering technique, and a derived extended clustering technique are explained in much details in (Barrière and Fass, 1998).

Table 1: Multiple clusters from different words

Trigger word	Cluster
needle_1	{needle_1, sew, cloth, thread, wool, handkerchief, pin, ribbon, string, rainbow}
sew	{sew, cloth, needle_1, needle_2, thread, button, patch_1, pin, pocket, wool, ribbon, rug, string, nest, prize, rainbow}
kitchen	{kitchen, stove, refrigerator, pan}
stove	{stove, pan, kitchen, refrigerator, pot, clay}
stomach	{stomach, kangaroo, pain, swallow, mouth}
airplane	{airplane, wing, airport, fly_2, helicopter, jet, kit, machine, pilot, plane}
elephant	{elephant, skin, trunk_1, ear, zoo, bark, leather, rhinoceros}
soap	{soap, dirt, mix, bath, bubble, suds, wash, boil, steam}
wash	{wash, soap, bath, bathroom, suds, bubble, boil, steam}

word in them.

3 Semantic disambiguation

We propose in this section a way to attempt at solving different types of semantic ambiguities by using the redundancy of information resulting from the clustering technique as briefly described in the previous section. Going through an example, we will look at three types of semantic ambiguity: anaphora resolution, word sense disambiguation, and relation disambiguation.

In Figure 1, Definition 3.1 shows one sentence in the definition of mail_1 (taken from the AHFD, as all other definitions in Figure 1) with its corresponding CG representation. Definition 3.2 shows one sentence in the definition of stamp also with its CG representation. Using the clustering technique briefly described in the previous section, the two words are put together into a cluster triggered by the concept [mail_1].

Result 3.1 shows the maximal join[3] between the two previous graphs around shared concept [mail_1]. Combining the information from stamp and mail_1, puts in evidence the redundant information. The reduction process for eliminating this redundancy will solve some ambiguities. This process is based on the idea of finding "compatible" concepts within a graph. Two concepts are compatible if their semantic distance is small. That distance is often based on

the relative positions of concepts within the concept hierarchy (Delugach, 1993; Foo et al., 1992; Resnik, 1995). For the present discussion we assume that two concepts are compatible if they share a semantically significant common supertype, or if one concept is a supertype of the other.

In Result 3.1, the concept [send] is present twice, and also the concept [letter] is present in two compatible forms: [letter] and [message]. The compatibility comes from the presence in the type hierarchy[4] of one sense of [letter], [letter_2], as being a subtype of [message]. These compatible forms actually allow the disambiguation of concept [letter] into [letter_2]. This should update the definition of stamp shown in Definition 3.2. The other sense of [letter], [letter_1] is a subtype of [symbol].

The pronoun they in Result 3.1 must refer to some word, either previously mentioned in the sentence, or assumed known (as a default) in the LKB. Both (agent) relations attached to concept [send] lead to compatible concepts: [they] and [person]. We can therefore go back to the graph definition of [stamp] in which the pronoun [they] could have referred to the concepts [letters], [packages], [people] or [stamps], and now disambiguate it to [people].

Result 3.2 shows the internal join which establishes coreference links (shown by *x, *y, *z) between compatible concepts that are in an identical relation with another concept. The reduced join, after the redundancy is eliminated, is shown in Result 3.3.

Two types of disambiguation (anaphora resolution and word sense disambiguation) were shown up to now. The third type of disambiguation is at the level of the semantic relations. For this type of ambiguity, we must briefly introduce the idea of a relation hierarchy which is described and justified in more details in (Barrière, 1998). A relation hierarchy, as presented in (Sowa, 1984), is simply a way to establish an order between the possible relations. The idea is to include relations that correspond to the English prepositions (it could be the prepositions of any language studied) at the top of the hierarchy, and consider them generalizations of possible deeper semantic relations.

---

[3]A maximal join is an operation defined within the CG formalism to gather knowledge from two graphs around a concept that they both share.

[4]The type hierarchy has been built automatically from information extracted from the AHFD.

Definition 3.1 -

MAIL_1 : People send messages through the mail.

[send]->(agent)->[person:plural]
    ->(object)->[message:plural]
    ->(through)->[mail_1]

Definition 3.2 -

STAMP : People buy stamps to put on letters and packages they send through the mail.

[send]->(object)->[letter:plural]->(and)->[package:plural]
                        <-(on)<-[put]<-(goal)<-[buy]->(agent)->[person:plural]
                                                ->(object)->[stamp:plural]
        ->(agent)->[they]
        ->(through)->[mail_1]

Result 3.1 - *Maximal join between mail_1 and stamp*

[send]->(object)->[letter:plural]->(and)->[package:plural]
                        <-(on)<-[put]<-(goal)<-[buy]->(agent)->[person:plural]
                                                ->(object)->[stamp:plural]
        ->(agent)->[they]
        ->(through)->[mail_1]<-(through)<-[send]->(object)->[message:plural]
                                                ->(agent)->[person:plural]

Result 3.2 - *Internal Join on Graph mail_1/stamp*

[send *y]->(object)->[letter:plural *z]->(and)->[package:plural]
                        <-(on)<-[put]<-(goal)<-[buy]->(agent)->[person:plural]
                                                ->(object)->[stamp:plural]
        ->(agent)->[they *x]
        ->(through)->[mail_1]<-(through)<-[send *y]->(object)->[message:plural *z]
                                                ->(agent)->[person:plural *x]

Result 3.3 - *After reduction of graph mail_1/stamp*

[letter_2:plural]->(and)->[package:plural]
        <-(object)<-[send]->(agent)->[person:plural]
                        ->(through)->[mail_1]
        <-(on)<-[put]<-(goal)<-[buy]->(agent)->[person:plural]
                                ->(object)->[stamp:plural]

Figure 1: Example of ambiguity reduction

106

Definition 3.3 -

CARD_2 : You send cards to people in the mail.

```
[send]->(agent)->[you]
      ->(object)->[card_2:plural]
      ->(to)->[person:plural]
      ->(in)->[mail_1]
```

Result 3.4 - *Graph mail_1/stamp joined to card (after internal join)*

```
[letter_2:plural]->(and)->[package:plural]
          <-(object)<-[send *y]->(agent)->[person:plural *z]
                         ->(through)->[mail_1]<-(in)<-[send *y]->(to)->[person:plural]
                                                              ->(agent)->[you *z]
                                                              ->(object)->[card_2:plural]

          <-(on)<-[put]<-(goal)<-[buy]->(agent)->[person:plural]
                                    ->(object)->[stamp:plural]
```

Result 3.5 - *After reduction of graph mail_1/stamp/card_2*

```
[letter_2:plural]->(and)->[package:plural]->(and)->[card_2:plural]
          <-(object)<-[send]->(agent)->[person:plural]
                         ->(manner)->[mail_1]
                         ->(to)->[person:plural]
          <-(on)<-[put]<-(goal)<-[buy]->(agent)->[person:plural]
                                    ->(object)->[stamp:plural]
```

Figure 1: Example of ambiguity reduction (continued)

This relation hierarchy is important for the comparison of graphs expressing similar ideas but using different sentence patterns that are reflected in the graphs by different prepositions becoming relations. Let us look in Figure 1 at Definition 3.3 which gives a sentence in the definition of [card_2] and Result 3.4 which gives the maximal join with graph mail_1/stamp from result 3.3 around concept [mail_1].

Subgraphs [send]->(in)->[mail_1] and [send]->(through)->[mail_1] have compatible concepts on both sides of two different relations. These two prepositions are both supertypes of a restricted set of semantic relations. On Figure 2 which shows a small part of the relation hierarchy, we highlighted the compatibility between **through** and **in**. It shows that the two prepositions interact at **manner** (at **location** as well but more indirectly). Therefore, we can establish the similarity of those two relations via the **manner** relation, and the ambiguity is resolved as shown in Result 3.5.

Note that the concept [person] is present many time among the different graphs in Figure 1. This gives the reader an insight into the complexity behind clustering. It all relies on compatibility of concepts and relations. Compatibility of concepts alone might be sufficient if the concepts are highly semantically significant, but for general concepts like [person], [place], [animal] we cannot assume so. In the graph presented in Result 3.5, there are buyers of stamps, receivers and senders of letters and they are all people, but not necessarily the same ones.

We saw the redundancy resulting from the clustering process and how to exploit this redundancy for semantic disambiguation. We see how redundancy at the concept level without the relations can be very misleading, and the following section emphasize the importance of semantic relations.
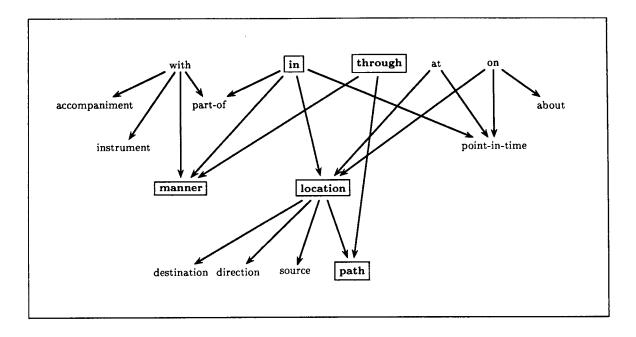
107

Figure 2: Small part of relation taxonomy.

4 The importance of semantic relations

Clusters are and have been used in different applications for information retrieval and word sense disambiguation. Clustering can be done statistically by analyzing text corpora (Wilks et al., 1989; Brown et al., 1992; Pereira et al., 1995) and usually results in a set of words or word senses. In this paper, we are using the clustering method used in (Barrière and Popowich, 1996) to present our view on redundancy and disambiguation. The clustering brings together a set of words but also builds a CCKG which shows the actual links (semantic relations) between the members of the cluster. We suggest that those links are essential in analyzing and disambiguating texts. When links are redundant in a graph (that is we find two identical links between two compatible concepts at each end) we are able to reduce semantic ambiguity relating to anaphora and word sense. The counterpart to this, is that redundancy at the concept level allows us to disambiguate the semantic relations.

To show our argument of the importance of links, we present an example. Example 4.1 shows a situation where an ambiguous word chicken (sense 1 for the animal and sense 2

for the meat) is used in a graph and needs to be disambiguated. If two graphs stored in a LKB contain the word chicken in a disambiguated form they can help solving the ambiguity. In Example 4.1, Graph 4.1 and Graph 4.2 have two isolated concepts in common: eat and chicken. Graph 4.1 and Graph 4.3 have the same two concepts in common, but the addition of a compatible relation, creating the common subgraph [eat]->(object)->[chicken], makes them more similar. The different relations between words have a large impact on the meaning of a sentence. In Graph 4.1, the word chicken can be disambiguated to chicken_2.

Example 4.1 -

John eats chicken with a fork.

Graph 4.1 -

[eat]->(agent)->[John]
    ->(with)->[fork]
    ->(object)->[chicken]

John's chicken eats grain.

Graph 4.2 -

[eat]->(agent)->[chicken_1]<-(poss)<-[John]
    ->(object)->[grain]

108

John likes to eat chicken at noon.

Graph 4.3 -

```
[like]->(agent)->[John]
      ->(goal)->[eat]->(object)->[chicken_2]
                     ->(time)->[noon]
```

Only if we look at the relations between words can we understand how different each statement is. It's all in the links... Of course those links might not be necessary at all levels of text analysis. If we try to cluster documents based on keywords, well we don't need to go to such a deep level of understanding. But when we are analyzing one text and trying to understand the meaning it conveys, we are probably within a narrow domain and the relations between words take all their importance. For example, if we are trying to disambiguate the word baseball (the sport or the ball), both senses of the words will occur in the same context, therefore using clusters of words that identify a context will not allow us to disambiguate between both senses. On the other hand, having a CCKG showing the relations between the baseball_1 (ball), the bat, the player and the baseball_2 (sport), will express the desired information.

5 Conclusion

We presented the problem of semantic disambiguation as solving ambiguities at the concept level (word sense and anaphora) but also at the link level (the relations between concepts). We showed that when gathering information around a particular subject via a clustering method, we tend to cumulate similar facts expressed in slightly different ways. That redundancy is expressed by multiple copies of compatible/identical concepts and relations in the resulting graph which is called a CCKG (Concept Clustering Knowledge Graph). The redundancy within the links (relations) helps disambiguate the concepts they connect and the redundancy within the concepts helps disambiguate the links connecting them. Clustering has been used a lot in previous research but only at the concept level; we propose that it is essential to understand the links between the concepts in the cluster if we want to disambiguate between elements that share a similar context of usage.

References

C. Barrière and D. Fass. 1998. Dictionary validation through a clustering technique. To be published in the Proceedings of Euralex'98: Eight EURALEX International Congress on Lexicography, Belgium, August 1998.

C. Barrière and F. Popowich. 1996. Concept clustering and knowledge integration from a children's dictionary. In *Proc. of the 16 th COLING*, Copenhagen, Danemark, August.

C. Barrière. 1997. *From a Children's First Dictionary to a Lexical Knowledge Base of Conceptual Graphs.* Ph.D. thesis, Simon Fraser University, June.

C. Barrière. 1998. The relation hierarchy: one key to representing natural language using conceptual graphs. Submitted at ICCS98: International Conference on Conceptual Structures, to be held in Montpellier, France, August 1998.

P. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai, and R.L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–480.

H. S. Delugach. 1993. An exploration into semantic distance. In H. D. Pfeiffer and T. E. Nagle, editors, *Conceptual Structures: Theory and Implementation*, pages 119–124. Springer, Berlin, Heidelberg.

N. Foo, B.J. Garner, A. Rao, and E. Tsui. 1992. Semantic distance in conceptual graphs. In T.E. Nagle, J.A. Nagle, L.L. Gerholz, and P.W.Eklund, editors, *Conceptual Structures: Current Research and Practice*, chapter 7, pages 149–154. Ellis Horwood.

F. Pereira, N. Tishby, and L. Lee. 1995. Distributional clustering of english words. In *Proc. of the 33 th ACL*, Cambridge,MA.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of the 14 th IJCAI*, volume 1, pages 448–453, Montreal, Canada.

J. Sowa. 1984. *Conceptual Structures in Mind and Machines.* Addison-Wesley.

Y. Wilks, D. Fass, G-M Guo, J. McDonald, T. Plate, and B. Slator. 1989. A tractable machine dictionary as a resource for computational semantics. In Bran Boguraev and Ted Briscoe, editors, *Computational Lexicography for Natural Language Processing*, chapter 9, pages 193–231. Longman Group UK Limited.