Lexical Chains for Question Answering

Dan Moldovan and **Adrian Novischi**
Computer Science Department
Richardson, TX 75083-O688 U.S.A.
moldovan@utdallas.edu

Abstract

The paper presents a method for finding topically related words on an extended WordNet. By exploiting the information in the WordNet glosses, the connectivity between the synsets is dramatically increased. Topical relations expressed as lexical chains on extended WordNet improve the performance of a question answering system by increasing the document retrieval recall and by providing the much needed axioms that link question keywords with answers.

1 Topical Relations and Lexical Chains

Topical relations are pointers that link a concept to other related concepts that may occur in a discourse. Topic changes from a discourse to another, and the same concept may be used in different contexts. When the context changes, the concepts that define the context change. For example, the game of tennis can be discussed from technical, regulations, or injuries points of view. In each case, the relevant concepts are quite different. This pluralism of topics for the same concept creates confusion and difficulty in identifying related concepts. Nevertheless, topical relations are useful for many applications such as: information retrieval, information extraction, text coherence, question answering, and others. They have been used extensively in computational linguistics to study: discourse, coherence, inference, implications, malapropisms, automatic creation of hypertext links, and others (Morris and Hirst 1991), (Harabagiu et al.1996) (Hirst and St-Onge 1998), (Green 1999), (Harabagiu and Moldovan 1998b).

Given a specific lexical knowledge base, such as WordNet (Fellbaum 1998), topical relations can be expressed as *lexical chains.* These are sequences of semantically related words that link two concepts. Current lexical chainers take advantage only of the WordNet relations between synsets, totally ignoring the glosses. A far better lexical chaining technology can be developed based on the information contained in the glosses of WordNet concepts. Consider the following text:

`S1: Jim was hungry.`
`S2: He opened the refrigerator.`

One can not explain the cohesion, and the intention of this simple text by using only the existing WordNet relations. This becomes surprisingly easy by observing that the glosses of {`hungry`}: `(feeling a need or desire to eat food)` and of {`refrigerator`}: `(a kitchen appliance in which food can be stored at low temperatures)` have in common the concept `food`. A chain can be constructed between `hungry` and `refrigerator` that explains the intention of opening the refrigerator, namely to get food, and thus demonstrating the cohesion of this text. Formal logic proofs can be derived when the glosses are transformed into logic forms. Consider we want to find the answer to the following question from the sentences in the example above:

Q: What did Jim want ?

The gloss of `want` is (`feel or have a desire`). This can be linked through lexical chains with the intention of sentences S1 and S2 and the following answer can be inferred:

A: Jim wanted to eat. (or Jim wanted food.)

2 Lexical Chains on Extended WordNet

2.1 Extended WordNet

In WordNet each concept (i.e synset) has a gloss that contains one or more definitions, comments and examples. These glosses contain words that are topically related to the words in the synsets. When a word in a gloss is semantically disambiguated, it points to the synset it belongs to.

One goal in the Extended WordNet project (Harabagiu, Miller and Moldovan 1999) is to semantically disambiguate all the glosses and map them to their corresponding synsets, thus increasing the connectivity between synsets. This allows the retrieval of topically related concepts. The derivation of topical relations is possible only because the glosses are semantically disambiguated.

2.2 Topical relations as paths between synsets

In WordNet, each concept is surrounded by a micro-context consisting of its gloss, the glosses of the concepts in the gloss, and the immediately connected other concepts. We extract from this micro-context the most representative concepts and form a unified list of topical relations. Then, for any two synsets S_i and S_j, we provide a mechanism to determine the paths that link the two concepts via topical relations, if such paths exist. This way, we can talk about the topical relations of a concept from the point of view of another concept. Naturally, this idea can be extended to more than two concepts.

The sources for the topical relations are illustrated in Figure 1.

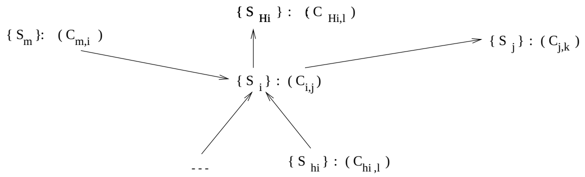

Figure 1: Topical relations are extracted from several sources

The first place where we look is the gloss of each concept. Since the gloss concepts are used to define a synset, they clearly are related to that synset. From the gloss definition we extract all the nouns, verbs, adjectives and adverbs, less some idiomatic expressions and general concepts that are not specific to that synset. Each concept $C_{i,j}$ in the gloss of synset S_i points to a synset S_j that has its own gloss definition. The concepts $C_{j,k}$ may also be relevant to the original synset S_i. Although this mechanism can be extended further, we will stop at this level.

A third source is the hypernym S_{Hi} and its gloss with concepts $C_{Hi,l}$. There is no need to include here the hyponyms of S_i as their topical list will point to S_i. Other relations such as causation, entailment and meronymy are treated in the same way as the hypernymy.

Yet another source consists of the glosses in which synset S_i is used. Those synsets, denoted in the figure with S_m are likely to be related to S_i since this is part of their definitions.

This scheme of constructing topical relations connects a synset to many other related synsets, some from the same part of speech hierarchy, but many from other part of speech hierarchies. The increased connectivity between hierarchies is an important addition to WordNet. Moreover, the creation of topical relations is a partial solution for the derivational morphology problem that is quite severe in WordNet.

Figure 2 shows five synsets S_1 to S_5, each with its topical relation list represented as a vertical line. In the list of synset S_i there are relations r_{ij} pointing to S_j.

It is possible to establish some connections between synsets via topical relations. We developed software that automatically provides connecting paths between any two synsets S_i and S_j up to a certain distance as shown in Table 1. The meaning of these paths is that the concepts along a path are topically related. These paths should not be confused with semantic distance or semantic similarity paths that can also be established on knowledge bases by using different techniques.

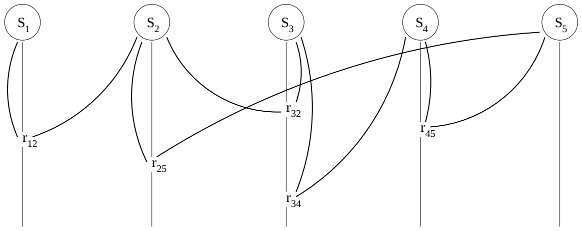

Figure 2: Topical relations for five synsets and the paths that can be established between the synsets

The name of the paths was inspired by their shape in Figure 2. The V path links directly S_i and S_j either via a $r_{i,j}$ or $r_{j,i}$, namely either S_i has S_j in its list of topical relations or vice versa. The W path allows one other interme-

Name	Path
V - path	$S_i - S_j$
W - path	$S_i - S_k - S_j$
VW - path	$S_i - S_k - S_l - S_j$
WW - path	$S_i - S_k - S_l - S_m - S_j$

Table 1: Types of topically related paths

diate synset between S_i and S_j. Similarly, the VW path allows two intermediate synsets and the WW path allows three intermediate synsets. We limit the length of the path to five synsets. For each type of path, there are several possible connections depending on the direction of the connection between two adjacent synsets (i.e. via $r_{i,j}$ or $r_{j,i}$). These connections are easily established with known search methods.

Examples

Suppose one wants to find out whether or not there are any topical relations between two words. The system will try to establish topical paths between any combination of the senses of the two words. Here are some examples:

Example 1: Is there any topical relation between `tennis` and `play`? (that is sense #1 of verb `play`)
Answer: there is a V-path: `play`:v#1 ← `tennis`:n#1
Explanation: `play`:v#1 is part of the definition of synset `tennis`:n#1, thus it is on its topical relations list.

Example 2: Is there any topical relation between `net` and `shuttlecock`?
Answer: there is a W-path: `net`:n#5 → `badminton`:n#4 ← `shuttlecock`:n#1
Explanation: `badminton`:n#1 is found in the definition of `net`:n#5 and is also found in the definition of `shuttlecock`:n#1.

Example 3: Is there a topical connection between `badminton` and `cork`?
Answer: there is a VW-path: `badminton`:n#1 → `racket`:n#4 → `shuttlecock`:n#1 → `cork`:n#1
Explanation: `badminton`:n#1 has `racket`:n#4 in its definition, which in turn has `shuttlecock`:n#1 in its definition, which is defined using `cork`:n#1 (i.e. a ball of cork or rubber...).

Example 4: Is there a possible topical relation between `play` and `cork`?
Answer: there is a WW-path: `play`:v#1 ← `badminton`:n#1 → `racket`:n#4 → `shuttlecock`:n#1 → `cork`:n#1
Explanation: `badminton`:n#1 has `play`:v#1 in its definition, and the rest is as before.

3 Finding Lexical Chains

3.1 Results on semantic disambiguation of WordNet glosses

The word sense disambiguation method comprises three complimentary components: (a) heuristics, (b) conceptual density and (c) statistics on large corpora. The accuracy of the three methods decreases from (a) to (c), while their generality increases. More details in (Harabagiu, Miller and Moldovan 1999).

An important feature of our WSD method is the capability to trade-off between the coverage (i.e. the number of words disambiguated) and the accuracy. When we disambiguate the entire set of WordNet glosses we obtain 100% coverage with 71% accuracy. The accuracy may be improved to 84% at the expense of dropping the coverage to 64% of the words.

3.2 Path finding procedures

Relations

Lexical chains are established along some selected WordNet relations (Harabagiu and Moldovan 1998a). However, not all relations are equal. To each relation we assign a weight between 0 and 1. The relations and their weights were determined empirically and are shown in Table 2. The first 13 relations are WordNet relations and are defined in the WordNet literature. The GLOSS relation is between a synset and the words in its gloss, and R-GLOSS between the gloss words and the respective synset.

Ranking the paths

When ranking the paths the following heuristics are used:

1. *Shorter paths are generally better than longer paths.*
2. *Relations are not equal.* They are ranked in Table 2.
3. *Order of relations in a path is important.*

Source concept	Reached concepts path size = 1	Reached concepts path size = 2	Reached concepts path size = 3
(birthing:a#1)	5	2324	24808
(expectant:a#2)	44	757	49512
(stably:r#1)	6	2533	22949
(population:n#1)	23	1884	45271
(aspirate:v#1)	3	395	6257
(like:v#1)	14	4173	50009

Table 3: Examples of destination concepts that can be reached starting from one source concept

Concept 1	Concept2	Paths number length = 1	Paths number length = 2	Paths number length = 3
(mother:n#1)	(woman:n#1)	1	12	95
(expectant:a#2)	(pregnant:a#1)	1	2	12
(population:n#1)	(people:n#1)	2	2	12
(originate:v#1)	(ancient:a#1)	0	1	9
(life:n#5)	(age:n#5)	1	2	27
(college:n#2)	(university:n#3)	2	10	26
(ancient:a#1)	(civilisation:n#1)	0	8	56

Table 4: The number of paths between pair of concepts

Relation	Weight
HYPERNYM	08
HYPONYM	0.7
ENTAIL	0.7
SIMILAR	0.9
IS-MEMBER-OF	0.5
IS-STUFF-OF	0.5
IS-PART-OF	0.5
HAS-MEMBER	0.5
HAS-STUFF	0.5
HAS-PART	0.5
CAUSE-TO	0.5
SEE-ALSO	0.5
PERTAIN	0.5
GLOSS	0.6
RGLOSS	0.2

Table 2: The weight assigned to each relation

4. *The type of nodes along paths is important.*

The rational for heuristics 1 and 2 is obvious. Regarding heuristic 3, consider the following paths:
(C1) → R-GLOSS → (C2) → GLOSS (C3) and
(C4) → GLOSS → (C5) → R-GLOSS (C6)
C1 and C3 are in the same gloss while C4 and C6 are related only through C5. The first path is stronger than the second one, and this illustrates the importance of ordering relations.

The rational for heuristic 4 is also obvious as the paths that pass through common concepts, like (have:v#1), are weaker than paths that contain more specific nodes.

The number of paths between two concepts can be a clue of how related two concepts are; the more paths the stronger the relation between the concepts.

Finding related concepts

An algorithm to automatically find concepts related to other concepts was developed. Given a set of concepts (taken from a question) we assign each concept a weight. For all other concepts in WordNet we assign value zero. Relations are weighted according to Table 2.

Each neighbour concept will receive a value which is the product between the weight of the original concept from the question and the weight of the relation. If we denote by W_G the weight of the original concept and W_R the relation weight taken for the above table, then the value of the concept at distance one is W_{D1}:

$W_{D1} = W_G * W_R$;

For example let's take the concept (mother:n#1). We assign to this concept the weight 10. Thus, concept (parent:n#1) which is a hypernym of concept (mother:n#1) will receive value 8. Concepts (mommy:n#1), (mother-in-law:n#1), (surrogate_mother#1), (Mary:n#1) which are hyponyms of mother will receive value 7. Concepts (woman:n#1), (give_birth:v#1), (child:n#1) will receive value 6. Concept (grandma:n#1) which has the concept (mother:n#1) in its gloss, will receive the value 2.

Then the concepts with the distance 2 from the original concept will receive a value which is a product of the original value of the first relation weight, the second relation weight, and a parameter to adjust the value of the two ordered relations. If we denote with W_{D2} the value of a concept at distance 2 from original concept, and W_{R1} and W_{R2} are the weights of the two relation, and $A_{1,2}$ is a parameter to adjust the ordering of relations R_1 and R_2 taken together, then:

$$W_{D2} = W_G * W_{R1} * W_{R2} * A_{1,2};$$

The parameter A12 is needed because the extent to which the two concepts are related depends directly on the relations weights, and also depends on the relation pair. For example to $A_{r-gloss,gloss}$ was assigned the value 3.0, but to $A_{gloss,r-gloss}$ was assigned the value 0.1. In the same way using the HYPERNYM and the HYPONYM relations, we consider that two brothers are closer related than two parents of the same concept (in different hierarchies). To $A_{hypernym,hyponym}$ we assigned the value 2.0, but to $A_{hyponym,hypernym}$ we assigned the value 0.8. For the rest of relation pairs $A_{i,j}$ we assigned the value 1.0. These values were found empirically, but they can be learned.

Generally, if a concept C' is found at a distance N from an original concept C and between the concept C and the concept C' there is a path that contains relations $R_1, R_2, ...R_N$, then the concept C' will receive the value:

$$W_{DN} = W_G * W_{R1} * W_{R2} * A_{1,2} * W_{R3} * A_{2,3} * ... * W_{RN} * A_{N-1,N};$$

We have to take care of the generality of a concept. For every concept we define the measure of generality MG_C which depends of the number of glosses in which a concept appear: $Nr - gloss$ is the number of R-GLOSS relations:

$$MG_C = \frac{CONST}{CONST+Nr-gloss};$$

For our algorithm we choose $CONST = 500$ but this value could be learned. If the path from C to C' is:

$$C \rightarrow R_1 \rightarrow C_1 \rightarrow R_2 \rightarrow C2 \rightarrow ... \rightarrow C_{N-1} \rightarrow R_N \rightarrow C_N = C',$$

the formula for computing the value of the destination concept C' becomes:

$$W_{DN} = W_G * W_{R1} * MG_{C1} * W_{R2} * A_{1,2} * MG_{C2} * ... * W_{RN} * A_{N-1,N} * MG_{CN};$$

Since multiple paths may intersect at one concept, that concept will receive many values. The total value is the sum of the values on the incoming paths. For example the concept (parent:n#1) is the hypernym of concept (mother:n#1), so it will receive a value of 8, but the concept (mother:n#1) also appears in its gloss so between (mother:n#1) and (parent:n#1) there is a R-GLOSS relation. Thus the concept (parent:n#1) will also receive the value 2. We add the two received values and get a value of 10 for the concept parent. This way, the number of paths between the two concepts determine how related two concepts are. The more paths, the more related the two concept are since the values add up.

Loops may be encounter. To avoid them, we create an exception list of concepts, and for any concept in this list do not compute any value.

The recursive algorithm of spreading weights follows.

Algorithm SpreadWeights

Inputs:
the weights for all WordNet concepts;
the weight of source concept (WG);
the current path from the source concept (path);
exception list of concepts;
left distance (d).

Output:
updated weights for all WordNet concepts.

```
1. if ( $d < 0$ ) then return;
2. current_concept = last concept in the path
3. compute the current value for the current path following the formula:
   $W_{DN} = W_G * W_{R1} * MG_{C1} * W_{R2} * A_{1,2} * MG_{C2} * ... * W_{RN} * A_{N-1,N} * MG_{CN}$;
4. add this value to the current concept.
   $weight[current\_concept] + = W_{DN}$
5. For all neighbour concepts $i$ of current_concept that are not already in the path or in the exception_list do:
   $new\_path = path \cup current\_node$;
6. call SpreadWeights( weight[], WG, new_path, exception_list, $d - 1$ );
```

Finding paths between two concepts

Paths between two concepts can now be found by simply checking the presence of a concept among the concepts reached from an initial concept. Almost any two concepts in WordNet can be related by a path of size 4 (a path with 5 concepts and 4 relations). It was measured that on average one concept has 12 immediate neighbours. Table 3 shows examples of the number of paths as a function of path size.

4 Applications to Question Answering

The design of Question Answering (QA) systems has long been a goal in artificial intelligence. In 1999, NIST introduced a new QA task in the Text REtrieval Conference (TREC) *http://trec.nist.gov/presentations/*. Our current QA system (Moldovan 1999) relies on searching a collection of documents using queries with question keywords, followed by answer extraction. Only questions that do not require complex reasoning can be answered with the current technology.

A major problem in QA is when the answer occurs in a form different from the question keywords. Lexical chains can help in two ways: (1) increase the recall of document retrieval by extending the keyword search with topically related words, and (2) improve the answer extraction by providing world knowledge axioms derived from the lexical chains linking question keywords with answer concepts.

Consider the question:

Q: How much folic acid should an expectant mother get daily?

A: ... Also, the term ” good source ” is based on the RDA of 400 micrograms daily for a pregnant woman.

The paragraph where the answer is does not contain the words ”mother” or ”expectant” but ”woman” and ”pregnant”. The concept (woman:n#1) is in the gloss of concept mother:n#1: (a woman who has given birth to a child). Also between concepts expectant:a#2 and pregnant:a#1 there is a SIMILAR relation. Using the words ”mother” and ”pregnant” the document containing the right answer is retrieved.

The lexical chains also help finding the exact answer in the document. For example:

Q: What day and month did John Lennon die ?

A: Similarly, former Beatle John Lennon was slain Dec. 8, 1980 by a deranged fan outside his New York apartment, ...

This answer is found only by linking the verb *die* with the verb *slain* . A lexical chain between concepts die:v#1 and slain:v#1 is:

`died:v#1` → R-GLOSS → `kill:v#1` → R-GLOSS `slain:v#1`

From the text containing the answer we infer that:

A1. John Lennon was killed Dec. 8 by a deranged fan ...

A2. John Lennon was caused to die Dec. 8 ...

A3. John Lennon die Dec. 8 ...

This last statement is the direct answer of the question. Finding the related concepts of a given concepts is the first step to inference which is performed with a logic prover.

Also the related words from the concepts of the question can improve the performance of question answering system.

Supplying the question answering system with the capability of deriving on-line lexical chains significantly improves the system performance. We took 100 questions that were not correctly answered by our system, but for which the correct answers (supplied by NIST) contain topically related words. Using the Spread Weights algorithm we checked if the missing words that caused the system failure appear in the top 5, top 10, or top 20 of the topically related words provided by the lexical chainer. The results are in the Table 5.

top set words	alternations found
top 5	11.2%
top 10	14.4%
top 20	33.8%

Table 5: Examples of alternation found with SpreadWeights algorithm

The axiomatic knowledge that results from the glosses is far from representing a rich world knowledge base. Nevertheless, new axioms can be derived with standard AI reasoning techniques by chaining several glosses.

References

Christiane Fellbaum. WordNet - *An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998.

S.M. Harabagiu, D.I. Moldovan and T.Yukawa. Testing Gricean constraints on a WordNet-based Coherence Evaluation System. In *Working Notes of the AAAI-96 Spring Symposium on Computational Approaches to Interpreting and Generating Conversational Implicature*, pages 31–38, 1996.

S.J. Green. Building Hypertext Links by Computing Semantic Similarity. *IEEE Tr. on Knowledge and Data Engineering*, pp 713-730, Sept/Oct 1999.

S.M. Harabagiu and D.I. Moldovan. A Parallel Inference System. *IEEE Transaction on Parallel and Distributed Systems*, Aug. 1998, pp 729-747.

S.M. Harabagiu and D.I. Moldovan. Knowledge Processing on an Extended WordNet. WordNet-An Electronic Lexical Database. The *MIT Press*, C. Fellbaum editor, pp 379-406, 1998.

S. Harabagiu, G.A. Miller, and D.I. Moldovan. WordNet 2 - A Morphologically and Semantically Enhanced Resource. *Proceedings of ACL-SIGLEX99: Standardizing Lexical Resources*, Maryland, June 1999, pp.1-8.

G. Hirst and D. St-Onge. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. WordNet-An Electronic Lexical Database. The *MIT Press*, C. Fellbaum editor, pp 305-332, 1998.

Dan Moldovan et al. Lasso: A tool for surfing the answer net. In *Proceedings of TREC-8, NIST 1999*, pp.65-73.

J. Morris and G. Hirst. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. In *Computational Linguistics*, Vol. 17 , no 1, pp. 21-48, 1991.