Lexical Chains on WordNet and Extensions

Tatiana Erekhinskaya and Dan Moldovan
Human Language Technology Research Institute
University of Texas at Dallas
Department of Computer Science University of Texas at Dallas
Richardson, TX

Abstract

Lexical chains between two concepts are sequences of semantically related words interconnected via semantic relations. This paper presents a new approach for the automatic construction of lexical chains on knowledge bases. Experiments were performed building lexical chains on WordNet, Extended WordNet, and Extended WordNet Knowledge Base. The research addresses the problems of lexical chains ranking and labeling them with appropriate semantic names.

Introduction to Lexical Chains

Lexical chains are sequences of semantically related words interconnected via semantic relations. They establish semantic connectivity between two end concepts. Lexical chains are constructed on knowledge bases that contain concepts and relations between concepts.

In this paper, lexical chains are built on three resources: WordNet (WN), eXtended WordNet (XWN), and eXtended WordNet Knowledge Base (XWN KB). Each of these resources can be viewed as a large semantic graph. Finding lexical chains consists in finding paths between concepts. In general, there are many possible chains between two concepts. For example, for pair $person - teach$ there are at least two useful lexical chains, giving different interpretations of connectivity:

1. $person : n\#1 \xrightarrow{\text{ISA}^{-1}} enrollee : n\#1 \xrightarrow{\text{ISA}^{-1}} student : n\#1 \xrightarrow{\text{DERIVATION}} educate : n\#1 \xrightarrow{\text{DERIVATION}} education : n\#1 \xrightarrow{\text{DERIVATION}} teach : v\#1$, where a person is the beneficiary of teaching, and
2. $person : n\#1 \xrightarrow{\text{ISA}^{-1}} leader : n\#1 \xrightarrow{\text{ISA}^{-1}} trainer : n\#1 \xrightarrow{\text{DERIVATION}} train : v\#1 \xrightarrow{\text{DERIVATION}} education : n\#1 \xrightarrow{\text{DERIVATION}} teach : v\#1$, where a person is doing the teaching.

ISA^{-1} (ie hyponymy) and DERIVATION are WordNet relations.

There are also meaningless chains which need to be filtered out by the system since they do not provide any useful semantic information. For example, *time:n#2* $\rightarrow$ *clock:v#1* $\rightarrow$ *certain:a#1* $\rightarrow$ *stand_by:v#2* $\rightarrow$ *wait:v#1*.

So, the task is to rank lexical chains and find the best one. In addition to finding the best chain overall, identifying all valid chains between two concepts may be of interest for some applications.

Related Work

Moldovan and Novischi (2002) proposed an algorithm for building lexical chains to find all related concepts with a given concept, and suggested that relatedness between pair of concepts can be checked by searching for a second concept in related concepts of the first one.

The authors also enhanced lexical chains with predicate-argument structures and used the structures for matching phrases (Novischi 2005). The proposed algorithm is based on transformation rules that modify the structure when propagated along the lexical chains. For example, there is a CAUSE relation between $feed : v\#2$ and $eat : v\#1$, the object of the verb $feed$ becomes the subject of eat. Using this rule, the sentence *"I fed a cat with fish."* can be matched with *"A cat ate fish."*

Lexical chains were successfully applied to Textual Entailment (Tatu and Moldovan 2006) and Question Answering (Harabagiu et al. 2005) to match similar concepts expressed with different words in different texts. A similar approach for building paths on semantic networks is used for text relatedness measurements (Tsatsaronis, Varlamis, and Vazirgiannis 2010). In (Onyshkevych 1997) a similar ontological graph search was applied to representation of text in the context of Knowledge-Based Machine Translation.

There is a wide variety of works employing lexical chains to represent text cohesion. A clear distinction should be made between these works and lexical chains built on lexical databases described in the previous section. Lexical chains on databases can be viewed as a mechanism for extracting knowledge from knowledge bases rather than representing text. There is no constraint that nodes on the path correspond to words present in some text.

To make the distinction more clear, lets look at an example given in (Morris and Hirst 1991): *Mary spent three hours in the garden yesterday. She was digging potatoes.* This example offers two consequent sentences and the lexical chain

$garden \rightarrow digging$ reveals the semantic connectivity between them. All nodes of the chain are found in the text.

Building lexical chains between $garden$ and dig on a lexical database will result in the following chain: *garden:n#1* $\xrightarrow{\text{ISA}}$ *plot:n#2* $\xrightarrow{\text{ISA}}$ *area:n#1* $\xrightarrow{\text{PART\_WHOLE}^{-1}}$ *land:n#1* $\xrightarrow{\text{LOCATION}^{-1}}$ *dig:v#1*.

Lexical chains built on a lexical database can be used to express connections between pairs of words in a text.

Most of the researchers using lexical chains to represent text use a limited number of relations. The most commonly used relations are SYNONYMY, ISA and PART_WHOLE from WordNet (Ercan, Gonenc and Cicekli, Ilyas 2007; Gonzàlez and Fuentes 2009), some use antonymy (Barzilay and Elhadad 1997), or siblings (hyponyms of hypernyms, e.g. dog and wolf) (Galley and Mckeown 2003). Hirst sets constraints on patterns of allowable paths (Hirst, Graeme and St-Onge, David 1998). Some researchers use Roget's Thesaurus (Jarmasz and Szpakowicz 2003) or domain-specific thesaurus (Medelyan 2007) as sources of relations.

Lexical Resources

Since the very notion of lexical chains is based on the concept of semantic relations, the resource that provides these relations is very important. WordNet (Fellbaum, Christiane 1998) is a large lexical database of nouns, verbs, adjectives and adverbs grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked via lexical relations, most of them between synsets of the same part of speech. Synsets are interlinked by means of conceptual-semantic and lexical relations; we use 9 of them: ISA, PART WHOLE, ENTAILMENT, CAUSE, SIMILAR, DERIVATION, ANTONYM, SEE ALSO, PERTAINYM, as well as their inverse relations.

WordNet contains a lot of implicit knowledge. To extract this knowledge in explicit form, Harabagiu, Miller and Moldovan (1999) proposed to process glosses in three ways:

- Semantic disambiguation of content words in glosses
- Syntactic parsing of glosses
- Transformation of glosses into logic form

The resulting resource, eXtended WordNet (XWN), makes possible the addition of a new relation, called GLOSS, between a synset and any of the concepts in its gloss. We denote GLOSS(x, y) the link between a synset x and a concept y in its gloss description. The addition of GLOSS relations helps increase the connectivity between concepts, and is based on the intuition that the concepts defining a synset have something to do with that synset.

For example, in WN, the concept *weatherman:n#1* is a hyponym of *meteorologist:n#1*. It has the gloss *"predicts the weather"*, so XWN allows to connect weatherman with *predicts:v#1* and *weather:n#1* via the GLOSS relation; i.e. GLOSS(*weatherman:n#1, predicts:v#1*) and GLOSS(*weatherman:n#1, weather:n#1*).

While eXtended WordNet helps to dramatically increase semantic connectivity between synsets, GLOSS relation is highly ambiguous and doesn't provide a valid semantic interpretation. For example, the concept $notation : n\#1$ has gloss *"a technical system of symbols used to represent special things"*. For example, it is obvious that the GLOSS relation between $notation$ and $system$ is stronger than between $notation$ and $special$.

A further step to enhance semantic connectivity of XWN concepts was taken in building the eXtended WordNet Knowledge Base. It is a new lexical resource where WordNet glosses are transformed into semantic graphs using semantic parser Polaris (Moldovan and Blanco 2012). The parser uses a fixed set of dyadic relations instead of uncontrollable large number of predicates with a variable number of arguments, some of relations are: AGENT, THEME, POSSESSION, LOCATION, MANNER, INSTRUMENT.

Using XWN KB, it is possible to replace the GLOSS relations with chains. For example, the two GLOSS relations above are replaced by a semantically more informative sequence: $weatherman : n\#1$ $\xrightarrow{\text{AGENT}}$ *predict:v#1* $\xrightarrow{\text{TOPIC}}$ *weather:v#1*.

In another example, the gloss of *tennis_player:n#1* is *"an athlete who plays tennis"*. There is a GLOSS relation between *tennis_player* and *tennis*, which can be replaced by the chain: *tennis_player:n#1* $\xrightarrow{\text{ISA}}$ *athlete:n#1* $\xrightarrow{\text{AGENT}}$ *play:v#1* $\xrightarrow{\text{THEME}^{-1}}$ *tennis:n#1*. If a GLOSS relation can be replaced with lexical chain $S_0 \xrightarrow{R_1} S_1 \rightarrow ... \rightarrow S_{n-1} \xrightarrow{R_n} S_n$, the corresponding GLOSS^{-1} relation can be replaced with the inverted chain defined as $S_n \xrightarrow{R_n^{-1}} S_{n-1} \rightarrow ... \rightarrow S_1 \xrightarrow{R_1^{-1}} S_0$.

Algorithm Constructing Lexical Chains

Extended WordNet Knowledge Base can be viewed as a large graph, and the task of building lexical chains consists of finding paths. A bidirectional search is used (Russell and Norvig 2010). So that the search tree is constructed for both source and target synsets, for each tree the breadth-first strategy is used. On each step, the tree with the smallest frontier (set of leaf nodes) is expanded.

Bidirectional search is applicable to the task, since for each relation in KB, it also maintains reverse relation, so that direct and backward search give the same results. The algorithm for building lexical chains is given below.

$ExpandLevel$ function consists of looping once over a given frontier and replacing each node with nodes reachable from it, except nodes in the path from root to the node. Optimization is done for frontier creation: $ExpandLevel$ function creates a list of newly created nodes for each expanded node, and then merges these lists together.

Weighting Lexical Chains

Weight Properties

In order to rank lexical chains, we need to assign a weight to each chain produced and determine a threshold to separate valid chains from bad ones. In this paper, unlike some other approaches (Novischi 2005), the weaker the chain is, the larger the weight. There are several desired features a weight function ought to have in order to be useful:

Algorithm 1: Building All Possible Lexical Chains

```
input  : Source, Target – pair of concepts
output: LexicalChains – set of established lexical
        chains
begin
    LexicalChains = ∅
    TargetFrontier = {Target}
    SourceFrontier = {Source}
    for i ← 1 to MAXITER do
        if |SourceFrontier| < |TargetFrontier|
        then
            SourceFrontier =
            ExpandLevel(SourceFrontier)
        else
            TargetFrontier =
            ExpandLevel(TargetFrontier)
        Intersection = TargetFrontier ∩ SourceFrontier;
        for Node in Intersection do
            SourcePath = GetPathBeween(Source,
            Node)
            TargetPath = GetPathBeween(Target, Node)
            TargetPath = Invert(TargetPath)
            Chain = Concatenate(SorcePath,
            TargetPath)
            LexicalChains = LexicalChains ∪ {Chain}
    Return LexicalChains
```

- Shorter chains are semantically stronger than longer ones, thus their weights are smaller than for longer chains.
- Weights should be monotonic: $W(C) \leqslant W(C \circ R)$
- Weights of chains should be such that the inversion operation holds: $W(C_1) \leqslant W(C_2) \Rightarrow W(C_1{}^{-1}) \leqslant W(C_2{}^{-1})$

The last intuition follows from the fact that if C is the best chain between concepts S_1 and S_2, then C^{-1} should be the best chain between S_2 and S_1.

Notice that $W(C)$ does not have to be equal to $W(C^{-1})$, because the direction of relations is important. For example, ISA is stronger than ISA^{-1}; ie ISA(*math, science*) is stronger than ISA^{-1}(*science, math*), simply because there is only one hypernym, while there are several hyponyms.

Weight Formula

Lexical chains between synsets S (source) and T (target) can be represented as $S_0 \xrightarrow{R_1} S_1 \rightarrow ... \rightarrow S_{n-1} \xrightarrow{R_n} S_n$, where $S_0 = S$ and $S_n = T$.

The weight of such lexical chain C is defined as:

$$W(C) = \sum_{i=0}^{n} W_{R_i} + W_{GC} \cdot N_{GC} + Penalty(C),$$

where W_{R_i} are the weights of relations R_i which form the chain. Some relations carry a stronger semantic connectivity than other relations, for instance ISA is stronger than ENTAILMENT, which is stronger than PERTAIN.

W_{GC} is a penalty for lexical chains containing a sequence of relations from two glosses. It is used only with XWN KB. N_{GC} is the number of such combinations in a chain.

$Penalty(C)$ is introduced to further discriminate against chains that contain frequently used words that in general do not add much semantic information when are part of GLOSS relation. We have formed lists with such stopwords, and if a chain contains a GLOSS relation with one of these words, a penalty larger than the threshold is added to the chain weight thus eliminating that chain. The list of adverbs stopwords L_{adv} contains 44 adverbs *(usually, sometimes, very)* and others; the list of adjectives stopwords L_{adj} contains 12 adjectives *(certain, specific, other)*; the list of verb stopwords L_{verb} contains 27 verbs *(be, use, have)*; and the list of noun stopwords L_{noun} contains 14 nouns *(act, state)*.

We used several strategies:

1. Filter out chains with common adverbs.
2. Filter out chains with common adverbs and verbs from stopword lists.
3. Filter out stopwords of all four parts of speech.

Experiments showed that the third strategy gives the best results.

To filter out bad chains, a threshold T is used, such that a valid chain C satisfies the condition $W(C) \leqslant T$. The values of parameters and threshold were set empirically. The strongest relation SIMILAR has a unit weight $W_{\text{SIMILAR}} = 1$, while $W_{\text{ISA}} = 1.1$ and $W_{\text{ISA}^{-1}} = 1.3$. Weights of relations from WN and XWN KB are in the range of 1 to 4. Relations GLOSS and GLOSS^{-1} are much weaker; $W_{\text{GLOSS}} = 13$ and $W_{\text{GLOSS}^{-1}} = 25$. Threshold is set to 30.

Semantic Calculus and Lexical Chains Interpretation

Lexical chains can help to determine the semantic meaning of the connection between concepts. Such semantic labeling of lexical chains would be useful for the interpretation of noun compounds. Also, relations between concepts, inferred using XWN KB, can be used for semantic parsing of sentences as set of preliminary hypothesis and establishing semantic relations between sentences.

Consider for example the chain: *combination lock* $\xrightarrow{\text{ISA}}$ *lock* $\xrightarrow{\text{IS\_PART}}$ *door*, we can infer *combination lock* $\xrightarrow{\text{IS\_PART}}$ *door*. The task is to formalize this intuition. For some chains there is no single dominating relation, so that short chain of relations defines the meaning of connection. For instance, for pair *digger:n#1 – spade:n#2* lexical chain $digger : n\#1 \xrightarrow{\text{AGENT}} dig : v\#1 \xrightarrow{\text{INSTRUMENT}^{-1}} spade : n\#2$ explains connectivity and can't be reduced to one relation.

The hypothesis is that using the composition operation: $R_i(S_{i-1}, S_i) \circ R_{i+1}(S_i, S_{i+1}) = R'(S_{i-1}, S_{i+1})$ it is possible to reduce the number of relations in the chain and then, either use a single remaining relation or pick up the dominating relation among several remaining relations or interpret resulting lexical chain in some meaningful way. This section is based on the notion of composition of semantic relation and its properties are discussed in (Blanco, Cankaya, and Moldovan 2010).

For some types of relations, transitivity can be used: $R(x,y) \circ R(y,z) \rightarrow R(x,z)$. For lexical chains 13 transitive relations are used: CAUSE, JUSTIFICATION, INFLUENCE, PURPOSE, SOURCE, ASSOCIATION, KINSHIP, IS-A, PART-WHOLE, POSSESSION, SYNONYMY, LOCATION, TIME.

For other relations explicit rules of format $R_1 \circ R_2 = R_3$ are used. To construct rules, all possible operands of composition should be reviewed. To reduce the number of pairs to be reviewed, the following property was used: $(R_1 \circ R_2)^{-1} = R_2^{-1} \circ R_1^{-1}$ (Blanco, Cankaya, and Moldovan 2010). Using this, it is enough to review the following pairs:

1. $R_1 \circ R_2$;
2. $R_1^{-1} \circ R_2$;
3. $R_1 \circ R_2^{-1}$;

where R_1 and R_2 are direct relations and $R_1 \neq R_2$.

For this project a list of 269 rules were constructed. For example, the rule THM $\circ$ LOC $=$ LOC helps to reduce lexical chain $book : n\#1 \xrightarrow{\text{THM}} read : v\#1 \xrightarrow{\text{LOC}} library : n\#1$ into relation $book : n\#1 \xrightarrow{\text{LOC}} library : n\#1$.

If there is a pure DERIVATION relation between noun and verb, it can be merged with the previous of following relation: DERIVATION $\circ R = R$ and $R \circ$ DERIVATION $= R$. This helps to extrapolate relation of verb to noun, expressing corresponding event, for example *swim:v#1 – swimming:n#1*.

The algorithm of lexical chain processing is based on the associativity of composition of semantic relations:
$R_1 \circ R_2 \circ R_3 = (R_1 \circ R_2) \circ R_3 = R_1 \circ (R_2 \circ R_3)$
Composing semantic relations whenever possible helps reduce the length of the chain.

Algorithm 2: Finding Dominating Relation

input : Source, Target – pair of concepts
output: LexicalChains – set of established lexical chains
begin
1. Loop over all relations r_i in the chain
1.1. If r_i and r_{i+1} are of the same type r and transitive, merge them into one r relation
1.2. else if there is a rule $R_1 \circ R_2 = R_3$, so that $R_1 = r_i$ and R$_2$ = r_{i+1}, merge these relations into R_3
1.3. else if there is a rule $R_1 \circ R_2 = R_3$, so that R$_2$ = r_i^{-1} and R$_2$ = r_{i+1}^{-1}, merge these relations into R_3^{-1}
1.4. else i=i+1
2. If there is only one relation in the chain, it is dominating relation.
Return dominating relation

Experiments and Results

Data

To evaluate the algorithms proposed, 473 pairs of concepts were picked up. 353 pairs were from WordSimilarity-353 corpus, which contains pairs of nouns used for testing semantic similarity (203 pairs) and relatedness (252 pairs) measures, some pairs are in both datasets (Finkelstein et al. 2002). Other pairs were taken by the authors from the FOSS corpus, which data was gathered from 3rd-6th grade students in schools utilizing the Full Option Science System (Nielsen and Ward 2007). The corpus consists of questions, reference (correct) answers and students answers. Answers are in free form, so they give a variety of paraphrasing examples. It covers the following areas: Life Science, Physical Science and Technology, Earth and Space Science, Scientific Reasoning and Technology. Pairs were picked up from the same answer or from pair reference answer – student answer. Pairs of concepts with different part of speech were targeted, since they are more difficult to connect with WordNet than pairs of nouns, and therefore show the usefulness of lexical chains.

Filtering Strategy Comparison

For evaluation, a corpus of 100 concept pairs was used. The results of the evaluation are shown in Table 1. N denotes the average number of chains per pair found by the program, n denotes the average number of chains considered valid by the program for each pair. These valid chains were reviewed by human and m were found valid for a pair on average. RR is the average reciprocal rank of the best chain picked up by human in the list of chains validated by the machine. We compared filtering strategies discussed earlier. Table 1 shows that strategy 3 – filter out all four part of speech stopwords – provides the best results.

Pair	N	n	m	RR
3rd strategy	452	28.6	8.5	0.75
2rd strategy	452	33.3	8.5	0.63
1rd strategy	452	46.6	8.5	0.6
no filtering	452	46.8	8.5	0.6

Table 1: Average number of chains per pair of concepts.

While 8.5 valid chains per pair of concepts seems large, in reality when these words are used in some context (ie text) there are other constraints that will pick up the most meaningful chain.

Comparison of Lexical Resources

We also ran the algorithm with all optimizations on these 100 pairs to compare lexical resources. The experiment showed that the results obtained for XWN KB are also superior when compared to WN or XWN. The average number of valid chains per pair of concepts was 46 for XWN KB, 29 for XWN and only 20 for WordNet. Moreover, using only WordNet relations, it is not possible to connect 16 pairs, whereas the program using XWN KB successfully finds valid lexical chains. It is clear, that WordNet alone is not sufficient. The problem with XWN is the GLOSS relations, which carry a big weight, so that some actually valid chains are filtered out. For XWN KB there are good examples of well connected pairs such as *light:a – mass:n, look:v – design:n, compare:v – same:a, distance:n – away:r*. They can not be established with WordNet.

Algorithm Efficiency

Efficiency of the algorithm can be estimated using the number of expanded nodes. Both approaches were tested on the same 100 pairs, setting $MAXITER = 4$ to compromise recall and time. Version without optimization requires expansion of 36468 nodes per pair in average, whereas version with optimization only 29373 per pair. Moreover, pruning allows to increase $MAXITER$ to 6, without significant increase of processing time.

Dominating Relation

To test the approach, Algorithm 2 was implemented and applied to all valid chains for 100 pairs of concepts, 3450 chains overall. Overall 22 percents of relations were reduced via merging, 52 percents of chains were changed up to some degree by the algorithm, and 16 percents were reduced to one dominating relation. The results highly depend on the pair of concepts connected. The most common modified chains are sequences of the following format:

- ISA and ISA^{-1}, meaning that concepts are sibling in ISA-hierarchy, for example $news : n\#1 \xrightarrow{\text{ISA}} information$: $n\#1 \xrightarrow{\text{ISA}^{-1}} report$:$n\#2$ was reduced from $news$: $n\#1 \xrightarrow{\text{ISA}} information$:$n\#1 \xrightarrow{\text{ISA}^{-1}} news$:$n\#4 \xrightarrow{\text{ISA}^{-1}}$ $report$:$n\#2$;
- PART_WHOLE and PART_WHOLE^{-1}, like in $planet : n\#1 \xrightarrow{\text{PART\_WHOLE}^{-1}} solarsystem : n\#1 \xrightarrow{\text{PART\_WHOLE}} sun : n\#1$, derrived from $planet : n\#1 \xrightarrow{\text{ISA}^{-1}} outerplanet : n\#1 \xrightarrow{\text{PART\_WHOLE}^{-1}} solarsystem : n\#1 \xrightarrow{\text{PART\_WHOLE}} sun : n\#1$.

There are some surprisingly good examples in the results, for example the following long chain: $musician : n\#1 \xrightarrow{\text{ISA}^{-1}} pianist : n\#1 \xrightarrow{\text{PART\_WHOLE}} technique : n\#1 \xrightarrow{\text{ISA}^{-1}} pianism : n\#1 \xrightarrow{\text{ISA}} performance : n\#1 \xrightarrow{\text{ISA}^{-1}} concert : n\#1$ was reduced to: $musician : n\#1 \xrightarrow{\text{PART\_WHOLE}} concert : n\#1$.

Error Analysis

Quality of lexical chains depends on two factors: lexical resource providing relations and approach for building and ranking them. Each of these factors has shortcomings; some of the problems are:

- Relations of the same type have the same weight, but it doesn't always means that they are equally strong. For example, relation ISA between $living\ thing$ and $object$ is much weaker than between $pure\ mathematics$ and $math$, but both pairs can be connected with one ISA relation in WordNet. The hypothesis is that the closer a pair of concepts connected with ISA relation is to the root of hierarchy, the weaker the strength of relation is. However, other relations also have different strengths. For example, LOC between $garage$ and car is stronger than between car and $person$. It is not trivial to measure the strength of individual relations.
- Connection through words that don't give much information. For example, the chain $lend : n\#1 \xrightarrow{\text{ISA}} change : v\#2 \xrightarrow{\text{ISA}^{-1}} dry : v\#1$ looks legitimate for the program, but doesn't make much sense, because the word $change$ by itself doesn't provide much information, $what$ is changed is missing here. Such words need be identified and added to the list of stopwords.

In addition to errors in eXtended WordNet Knowledge Base, which are the result of mistakes of semantic parser, there are some disadvantages of WordNet:

- WordNet does not contain enough "marketing" information about brands, products and companies. For example, there are $Coca - Cola$, $Windows$ and $Java$ (programming language) in WordNet, but $Beetle$ or $Jaguar$ do not have senses related to car.
- WordNet does not contain common sense knowledge, it is non-trivial to connect cat and $mouse$ through $eating$; or establish THEME relation between $door$ and $open$. Another example is $popcorn$ and $movie$, naturally connected to the human via situation frame, but not that well connected in dictionary.
- Lack of specificity in glosses. For example, the word $right$ has gloss *"an abstract idea of that which is due to a person or governmental body by law or tradition or nature"*. It gives connection of $right$ and law, but the part *that what is due* is explicit, it would better to have something like *"what is due to do or possess"*.
- Granularity of WordNet senses can make meaningful lexical chains longer, more convoluted and harder to assign correct weight.
- Actual relation strength in WordNet is not uniform, for example, the deeper in ISA-hierarchy relation is, the stronger it is. Relation strength also depends on how many relations the concept has. Specific concepts have fewer relations and they are more important than relations of general words.

Conclusion and Future Work

The main conclusion of this paper is that valid lexical chains can be obtained using the XWN KB resource. It has the glosses of WN transformed into semantic relations by a semantic parser. WN lacks the connectivity to be useful for providing lexical chains, and XWN using GLOSS relations sometimes provides poor ranking. Since XWN KB is machine generated, without human validation, it contains errors and these were responsible for the mistakes we saw in some of the chains produced. Even XWN KB lacks the connectivity between some concepts and sometimes no valid chains can be produced. This was the case for the pairs *flight:n#2 – dinner:n#1*, and *predict:v#1 – volatility:n#2* in spite of human intuition that such chains exist.

Some relations in lexical chains are more important than others, because they convey the overall meaning of connectivity between concepts. To find such dominating relations, semantic calculus was used. An algorithm based on transformation rules was developed, allowing to combine several relation into one. The algorithm helps to significantly shorten lexical chains and make them more meaningful. For 16% of concept pairs it is possible to find a dominating relation for their lexical chains.

The most important future task is to automatically correct XWN KB using linear and syntax-based patterns and further enrich connectivity using other lexical resources. The mutual help of semantic parser and XWN KB also looks promising. Semantic parser can use relations given by current version of XWN KB to improve parsing of glosses and create new versions of XWN KB. The correctness of knowledge representation in XWN KB can also be improved by considering negations in glosses.

In addition to the notion of dominating relation, there is also the observation about central concept – most important node in the chain, explaining the relation between end concepts. For example, the concepts *tennis:n#1* and *person:n#1* are connected by a number of chains, some are:

1. *person:n#1* $\xrightarrow{\text{ISA}^{-1}}$ *athlete:n#2* $\xrightarrow{\text{ISA}^{-1}}$ *tennis_player:n#1* $\xrightarrow{\text{GLOSS}}$ *tennis:n#1*
2. *person:n#1* $\xrightarrow{\text{ISA}^{-1}}$ *official:n#2* $\xrightarrow{\text{ISA}^{-1}}$ *linesman:n#1* $\xrightarrow{\text{GLOSS}}$ *tennis:n#1*
3. *person:n#1* $\xrightarrow{\text{ISA}^{-1}}$ *coach:n#1* $\xrightarrow{\text{GLOSS}^{-1}}$ *sport:n#2* $\xrightarrow{\text{ISA}^{-1}}$ *professional_tennis:n#1* $\xrightarrow{\text{ISA}^{-1}}$ *tennis#1*

All three chains are valid, allowing different interpretations relating the two concepts – that a person in tennis can be player, linesman or coach.

References

Barzilay, R., and Elhadad, M. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, 10–17.

Blanco, E.; Cankaya, H. C.; and Moldovan, D. I. 2010. Composition of semantic relations: Model and applications. In Huang, C.-R., and Jurafsky, D., eds., *COLING (Posters)*, 72–80. Chinese Information Processing Society of China.

Ercan, Gonenc and Cicekli, Ilyas. 2007. Using lexical chains for keyword extraction. *Inf. Process. Manage.* 43(6):1705–1714.

Fellbaum, Christiane. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; and Ruppin, E. 2002. Placing search in context: the concept revisited. *ACM Trans. Inf. Syst.* 20(1):116–131.

Galley, M., and Mckeown, K. 2003. Improving word sense disambiguation in lexical chaining. In *Proceedings of IJCAI*, 1486–1488.

Gonzàlez, E., and Fuentes, M. 2009. A new lexical chain algorithm used for automatic summarization. In *Proceedings of the 2009 conference on Artificial Intelligence Research and Development: Proceedings of the 12th International Conference of the Catalan Association for Artificial Intelligence*, 329–338. Amsterdam, The Netherlands: IOS Press.

Harabagiu, S. M.; Moldovan, D. I.; Clark, C.; Bowden, M.; Hickl, A.; and Wang, P. 2005. Employing two question answering systems in trec 2005. In Voorhees, E. M., and Buckland, L. P., eds., *TREC*, volume Special Publication 500-266. National Institute of Standards and Technology (NIST).

Harabagiu, S. M.; Miller, G. A.; and Moldovan, D. I. 1999. WordNet 2 - A Morphologically and Semantically Enhanced Resource. In *Proceedings of SIGLEX*. Association for Computational Linguistics.

Hirst, Graeme and St-Onge, David. 1998. *Lexical Chains as representation of context for the detection and correction malapropisms*. The MIT Press.

Jarmasz, M., and Szpakowicz, S. 2003. S.: Roget thesaurus and semantic similarity. In *In: Proceedings of the RANLP-2003*, 212–219.

Medelyan, O. 2007. Computing lexical chains with graph clustering. In *Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop*, ACL '07, 85–90. Stroudsburg, PA, USA: Association for Computational Linguistics.

Moldovan, D., and Blanco, E. 2012. Polaris: Lymba's semantic parser. In Chair), N. C. C.; Choukri, K.; Declerck, T.; Dogan, M. U.; Maegaard, B.; Mariani, J.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA).

Moldovan, D., and Novischi, A. 2002. Lexical chains for question answering. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, 1–7. Morristown, NJ, USA: Association for Computational Linguistics.

Morris, J., and Hirst, G. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Comput. Linguist.* 17(1):21–48.

Nielsen, R. D., and Ward, W. 2007. A corpus of fine-grained entailment relations. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 28–35. Prague: Association for Computational Linguistics.

Novischi, A. 2005. *Semantic Disambiguation of Wordnet Glosses and Lexical Chains on Extended WordNet*. Ph.D. Dissertation, University of Texas at Dallas.

Onyshkevych, B. A. 1997. *An Ontological-Semantic Framework for Text Analysis*. Ph.D. Dissertation, Carnegie Mellon University.

Russell, S. J., and Norvig, P. 2010. *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education.

Tatu, M., and Moldovan, D. 2006. A logic-based semantic approach to recognizing textual entailment. In *Proceedings of the COLING/ACL on Main conference poster sessions*, 819–826.

Tsatsaronis, G.; Varlamis, I.; and Vazirgiannis, M. 2010. Text relatedness based on a word thesaurus. *Artificial Intelligence Research* 1–39.