The Look and Feel of a Confident Entailer

Vasile Rus[*], **Art Graesser**[†]

[*]Department of Computer Science
[†]Department of Psychology
Institute for Intelligent Systems
The University of Memphis
Memphis, TN 38152
{vrus, a-graesser}@memphis.edu

Abstract

The paper presents a software system that embodies a lexico-syntactic approach to the task of Textual Entailment. Although the approach is based on a minimal set of resources it is highly confident. The architecture of the system is open and can be easily expanded with more and deeper processing modules. Results on a standard data set are presented.

1. Introduction

Recognizing textual entailment (RTE) (Dagan et al., 2005) is the task of deciding, given two text fragments, whether the meaning of one text is entailed (can be inferred) from another text (Dagan et al., 2005). We say that T - the entailing text, entails H - the entailed hypothesis. The task is relevant to a large number of applications, including machine translation, question answering, and information retrieval.

We present a software system that implements an approach to RTE that uses minimal knowledge resources. The approach (Rus et al., 2005) only uses lexical, syntactic, synonymy and antonymy information. The synonymy and antonymy information is extracted from a thesaurus, an online dictionary. No deeper processing, word knowledge or automated reasoning is involved.

In our approach each T-H pair is first mapped into two graphs, one for H and one for T, with nodes representing main concepts and links indicating syntactic dependencies among concepts as encoded in H and T, respectively. An entailment score, $entail(T,H)$, is then computed quantifying the degree to which the T-graph subsumes the H-graph. The score is so defined to be non-reflexive, i.e. $entail(T,H) \neq entail(H,T)$. We evaluated the approach on the standard RTE challenge (Dagan et al., 2004 2005) data. The results obtained are promising.

2. Related Work

The task of textual entailment was treated in the recent past in one form or another by research groups ranging from informational retrieval to language processing. It is important to have a glimpse of what approaches have been adopted to better position our work. However, due to space limitations the approaches are briefly enumerated and not discussed in depth.

In one of the earliest explicit treatments of entailment Monz and de Rijke (Monz and de Rijke, 2001) proposed a weighted bag of words approach to entailment. Recently, Dagan and Glickman (Dagan and Glickman, 2004) presented a probabilistic approach to textual entailment based on lexico-syntactic structures. Pazienza and colleagues (Pazienza et al., 2005) use syntactic graph distance approach for the task of textual entailment. More recently, Kouylekov and Magnini (Kouylekov and Magnini, 2005) approached the entailment task with a tree edit distance algorithm on dependency trees.

A closely related effort is presented in (Moldovan and Rus, 2001). They show how to use unification and matching to address the answer correctness problem. Answer correctness can be viewed as entailment: Is a candidate answer entailing the ideal answer to the question? Initially, the question is paired with an answer from a list of candidate answers (obtained through some keyword proximity and shallow semantics methods). The resulting pair is mapped into a first-order logic representation and a unification process between the question and the answer that follows. As a back-off step, for the case when no full unification is possible, the answer with highest unification score is ranked at the top. The task they describe is different than the RTE task because a list of candidate answers to rank are available. The granularity of candidate answers and questions is similar to the RTE data.

3. Approach

Our solution for recognizing textual entailment is based on the idea of subsumption. In general, an object X subsumes an object Y if X is more general than or identical to Y, or alternatively we say Y is more specific than X. The same idea applies to more complex objects, such as structures of interrelated objects. Applied to textual entailment, subsumption translates into the following: hypothesis H is entailed from T if and only if T subsumes H.

4. The System Overview

The solution has two phases: (I) map both T and H into graph structures and (II) perform a subsumption operation between the T-graph and H-graph.

4.1. Phase I: From Text To Graph Representations

The two text fragments involved in a textual entailment decision are initially mapped into a graph representation that has its roots in the dependency-graph formalisms of (Mel'cuk, 1998). The mapping process has three phases: preprocessing, dependency graph generation and final graph generation.

$$entscore(T,H) = (\alpha \times \frac{\sum_{V_h \in H_V} max_{V_t \in T_V} match(V_h, V_t)}{|V_h|} + \beta \times \frac{\sum_{E_h \in H_E} max_{E_t \in T_E} synt\_match(E_h, E_t)}{|E_h|} + \gamma) \times \frac{(1 + (-1)^{\#neg\_rel})}{2} \quad (1)$$

In the preprocessing phase we perform tokenization (separation of punctuation from words), lemmatization (map morphological variations of words to their base or root form), part-of-speech tagging (assign parts of speech to each word) and parsing (discover major phrases and how they relate to each other, with the phrases being grouped into a parse tree). The preprocessing continues with a step in which parse trees are transformed in a way that helps the graph generation process in the next phase. For example, auxiliaries and passive voice are eliminated but their important information is kept: voices are marked as additional labels to the tag that identifies the verb; verb aspect information for the verb a modal (may, must, can) acts upon is recorded as an extra marker of the node in the graph that is generated for the verb. An important step, part of the preprocessing phase, identifies major concepts in the input: named entities, compound nouns and collocations, postmodifiers, existentials, etc. This step is important because, for instance, entities may appear as composed of multiple words in T, e.g. *Overture Services Inc*), and as a single word concept in H, e.g. (*Overture*). To assure a proper treatment of those cases only common collocations, namely those composed of a sequence of common nouns in the input, are represented as a single concept by replacing the consecutive words forming a collocation with a new concept composed of the individual words glued with an underscore. A dictionary of collocations (compiled from WordNet (Miller, 1995)) and a simple algorithm help us detect collocations in the input.

The actual mapping from text to the graph representation is based on information from parse trees. A parse tree groups words in a sentence into phrases and organizes phrases in hierarchical tree structures from where we can easily detect syntactic dependencies among concepts. We use Charniak's (Charniak, 2000) parser to obtain parse trees and head-detection rules (Magerman, 1994) to obtain the head of each phrase. A dependency tree is generated by linking the head of each phrase to its modifiers in a straightforward mapping step. The problem with the dependency tree is that it only encodes local dependencies (head-modifiers). Remote dependencies are not marked in such dependency trees. An extra step transforms the previous dependency tree into a dependency graph (Figure 1) in which remote dependencies are explicitly marked and further into a final graph in which direct relations among content words are coded. For instance, a *mod* dependency between a noun and its attached preposition is replaced by a direct dependency between the prepositional head and prepositional object.

The remote dependencies are obtained using a naive-Bayes functional tagger. The naive Bayesian model relies on more than a dozen linguistic features automatically extracted from parse trees (phrase label, head, part of speech, parent's head, parent's label, etc.). The model was trained on annotated data from Wall Street Journal section of Penn Treebank (Marcus et al., 1993). The accuracy of the functional tagger is in the 90-th percentile (Rus and Desai, 2005).
As soon as graph representations are obtained, a graph matching operation is initialized. The operation is detailed in the next section.

4.2. Phase II: Graph Subsumption

Let us remember core concepts from graph theory before we proceed with modelling subsumption for textual entailment.
A graph $G = (V, E)$ consists of a set of nodes or vertices V and a set of edges E. Isomorphism in graph theory is the problem of testing whether two graphs are really the same (Skiena, 1998). Several variations of graph isomorphism exist in practice of which the subsumption or containment problem best fits our task. Is graph H contained in (not identical to) graph T? Graph subsumption consists of finding a mapping from vertices in H to T such as edges among nodes in H hold among mapped edges in T. In our case the problem can be further relaxed: attempt a subsumption and if that is not possible back-off to a partial subsumption. The important aspect is to quantify the degree of subsumption of H by T.
The subsumption algorithm for textual entailment has three major steps: (1) find an isomorphism between H_V (set of vertices of the Hypothesis graph) and T_V (2) check whether the labelled edges in H, H_E, have correspondents in T_E (3) compute score. Step 1 is more than a simple word-matching method since if a node in H does not have a direct correspondent in T a thesaurus is used to find all possible synonyms for nodes in T. Nodes in H have different priorities: head words are most important followed by modifiers. Modifiers that indicate negation are handled separately from the bare lexico-syntactic subsumption since. If H is subsumed at large by T, and T is not negated but H is or viceversa, the overall score should be dropped, with high confidence, to indicate no entailment. Step 2 takes each relation in H and checks its presence in T. It is augmented with relation equivalences among appositions, possessives and linking verbs (*be, have*). Lastly, a normalized score for node and edge mapping is computed. The score for the entire entailment is the sum of each individual node and relation matching score. The node match consists of lexical matching and aspect matching (for verbs). The overall score is sanctioned by negation relations.

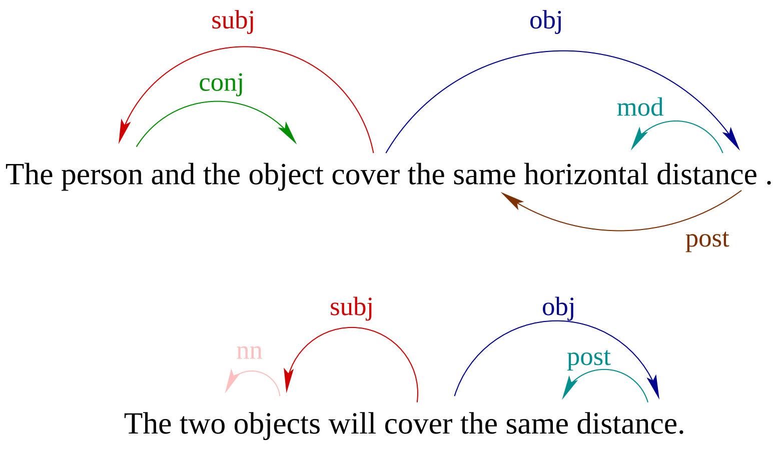
subj
obj
conj
mod
The person and the object cover the same horizontal distance .
post
subj
obj
nn
post
The two objects will cover the same distance.

Figure 1: Example of graph representation for a Text (top) - Hypothesis (bottom) pair.(Edges are colour-coded to better visualize the correspondence.)

4.3. Negation

We pay special attention to two broad types of negation: explicit and implicit. Explicit negation is indicated by particles such as: *no, not, neither ... nor* and their shortened forms *'nt*. Implicit negation is present in text via deeper lexico-semantic relations among different linguistic expressions, the most obvious example is the *antonymy* relation among lemmas which can be retrieved from WordNet. Negation is regarded as a feature of both text and hypothesis and it is accounted for in the score after the entailment decision for the Text-Hypothesis pair without negation is made. If one of the text fragments is negated the decision is reversed while if both are negated the decision is retained (double-negation), and so forth. In Equation 1 the term *#neg_rel* represents the number of negation relations between T and H.

5. The Structure of the Textual Entailer

The major subsystems of the textual entailer are the lexical matching, syntactic matching and negation (see Figure 2). Each subsystem is composed of several modules.
The lexical matching subsystem includes tokenization, lemmatization, collocations, part of speech tagging, and synonymy extraction component (from a thesaurus). The syntactic matching subsystem is composed of parsing, local dependency relations extraction and remote dependency relations extraction. The negation subsystem contains the explicit negation handling component and implicit negation (antonymy identification with WordNet) handling component.

6. Brief Summary of Results

The evaluation is automatic and follows the guidelines from RTE (Dagan et al., 2004 2005). The judgements (classifications) returned by the system are compared to those manually assigned by the human annotators (the gold standard). The percentage of matching judgements provides the accuracy of the run, i.e. the fraction of correct responses. A Confidence-Weighted Score (CWS, also known as average precision) is also computed. Judgments of pairs are sorted by their confidence (in decreasing order from the most certain to the least certain), calculating the following measure:

$$\frac{1}{n} * \sum_{i=1}^{n} \frac{\# - correct - up - to - pair - i}{i} \quad (2)$$

where n is the number of the pairs in the test set, and i ranges over the pairs.
The Confidence-Weighted Score varies from 0 (no correct judgements at all) to 1 (perfect score), and rewards the systems' ability to assign a higher confidence score to the correct judgements than to the wrong ones.
We used the development RTE data to estimate the parameters of the score equation and then applied the equation with the best found parameters to test data. We used linear regression to estimate the values of the parameters and also experimented with balanced weighting ($\alpha = \beta = 0.5$, $\gamma = 0$). The balanced weighted scheme provides better results and the performance figures are obtained with this scheme. The score provided by the formula is further used to find the entailment decision (TRUE or FALSE) and the level of confidence. Depending on the value of the overall score three levels of confidence are assigned: 1, 0.75, 0.5. For instance, an overall score of 0 leads to FALSE entailment with maximum confidence of 1. In summary we obtained an accuracy 0.554% of and a cws score of 0.604%. The results reported by our graph based approach on test data are significant at 0.01 level. Our cws is the highest as compared to approaches that use similar array of resources (see (Rus et al., 2005) for more comparison details). Since a good cws score indicates confidence we can claim our system is a confident entailer. This argument becomes stronger when one thinks of the limited array of resources we use.

7. Discussion, Further Work and Conclusions

The paper presents a software system that implements a lexico-syntactic approach to the task of textual entailment. Modules such as tokenization, lemmatization, tagging and parsing have been incrementally added to the system and more can be added on top of it. More, deeper modules can

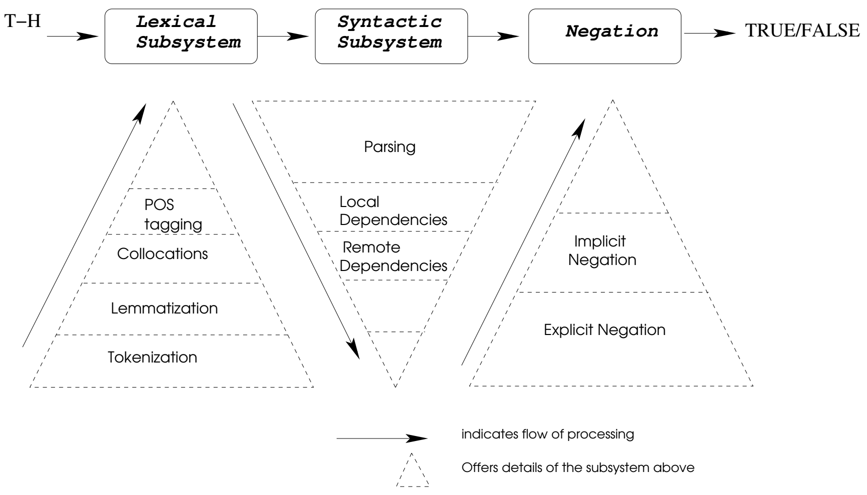

Figure 2: The Architecture of the Textual Entailer

be appended for enhanced performance. We plan to study the nature of interactions between different modules and what the contribution of each module to the overall performance of the system is. While the task at hand is difficult, in particular on the RTE data which is balanced (50% leads to FALSE and 50% leads to TRUE), and the impact of different modules can be small, we plan to study the contribution of each module on subsets of data form where we can draw better conclusions.

Furthermore, we are investigating the application of this approach to Intelligent Tutoring Systems (ITS), namely AutoTutor (Graesser et al., 2004). The example in Figure 1 is from an AutoTutor log file and represents a pair composed of an expectation, or ideal answer to a problem, and the student answer to the problem. The expectation-student answer is equivalent to the Text-Hypothesis pair and thus approaches to textual entailment can be transferred to evaluating student answers in ITS environments.

8. References

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of North American Chapter of Association for Computational Linguistics (NAACL-2000)*, Seattle, WA, April 29 - May 3.

I. Dagan and O. Glickman. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. In *Proceedings of Learning Methods for Text Understanding and Mining*, Grenoble, France, January 26 - 29.

I. Dagan, O. Glickman, and B. Magnini. 2004-2005. Recognizing textual entailment. In *http://www.pascal-network.org/Challenges/RTE*.

I. Dagan, O. Glickman, and B. Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the Recognizing Textual Entaiment Challenge Workshop*, Southampton, U.K., April 11 - 13.

A.C. Graesser, S. Lu, G.T. Jackson, Mitchell, H., M. Ventura, A. Olney, and M.M. Louwerse. 2004. Autotutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers*, pages 180–193.

M. Kouylekov and B. Magnini. 2005. Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the Recognizing Textual Entaiment Challenge Workshop*, Southampton, U.K., April 11 - 13.

D.M. Magerman. 1994. *Natural Language Parsing as Statistical Pattern Recognition*. Ph.D. thesis, Stanford University, February.

M. Marcus, B. Santorini, and Marcinkiewicz. 1993. Building a large annotated coprus of english: the penn treebank. *Computational Linguistic*, 19(2):313–330.

I.A. Mel'cuk. 1998. *Dependency Syntax: theory and practice*. State University of New York Press, Albany, NY.

George Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

D.I. Moldovan and V. Rus. 2001. Logic form transformation of wordnet and its applicability to question answering. In *Proceedings of the ACL Conference (ACL-2001)*, Toulouse, France, July.

C. Monz and M. de Rijke, 2001. *Light-Weight Entailment Checking for Computational Semantics*, pages 59–72.

M.T. Pazienza, M. Pennacchiotti, and F.M. Zanzotto. 2005. Textual entailment as syntactic graph distance: A rule based and svm based approach. In *Proceedings of the Recognizing Textual Entaiment Challenge Workshop*, Southampton, U.K., April 11 - 13.

V. Rus and K. Desai. 2005. Assigning function tags with a simple model. In *Proceedings of Conference on Intelligent Text Processing and Computational Linguistics (CICLing) 2005*, Mexico City, Mexico.

V. Rus, A. Graesser, and P.M. McCarthy. 2005. Lexico-syntactic subsumption for textual entailment. In *Lexico-Syntactic Subsumption for Textual Entailment*, volume Recent Advances in Natural Language Processing (RANLP), Borovets, Bulgaria, September 17-19.

S.S. Skiena. 1998. *The Algorithm Design Manual*. Springer-Verlag.