PROBABILISTIC TEXTUAL ENTAILMENT: GENERIC APPLIED MODELING OF LANGUAGE VARIABILITY

Ido Dagan, Oren Glickman
Computer Science Department
Bar Ilan University, Ramat Gan 52900, Israel
{dagan,glikmao}@cs.biu.ac.il

1 Introduction

A most prominent phenomenon of natural languages is variability – stating the same meaning in various ways. Robust language processing applications – like Information Retrieval (IR), Question Answering (QA), Information Extraction (IE), text summarization and machine translation – must recognize the different forms in which their inputs and requested outputs might be expressed. Today, inferences about language variability are often performed by practical systems at a "shallow" semantic level, due to the fact that robust semantic interpretation into logic-based meaning-level representations is not feasible. However, there is yet no generally applicable framework for modeling variability in an application independent manner. Consequently this problem is treated mostly independently within individual systems, and usually to a quite limited extent. In this paper we outline a proposal for a *generic* model for recognizing language variability at a shallow semantic level, its implementation as a practical engine to be leveraged within a variety of applications, and several learning tasks that it poses.

Our approach is based on a notion of textual entailment between text expressions, capturing that the meaning of one expression can be inferred from the other. We propose an inference model that approximates entailment without any explicit interpretation into meaning representations, but rather operating directly over lexical-syntactic units. The model consists of a knowledge base of basic patterns along with compositional probabilistic inference rules, and can be implemented as a practical Prolog-style engine. We further propose learning approaches for acquiring the required knowledge. We suggest that the proposed model may provide a unifying compositional framework for a broad range of shallow semantic inferences. As will be demonstrated below, articulating such a generic model leads to identifying various novel problem areas that need to be addressed in order to achieve complete inferences. We therefore believe that progress along this framework is likely to promote the levels of "understanding" and performance of multiple language processing applications.

2 Textual Entailment

We base our approach for recognizing language variability on modeling entailment between language expressions, specifying that the meaning of one expression can be inferred from the other. This relationship is directional since the meaning of one expression (e.g. "buy") may usually entail the other (e.g. "own") while entailment in the other direction is much less certain. While entailment pertains to the meaning of language expressions, the proposed model does not represent meanings explicitly, avoiding any semantic interpretation into a meaning representation level. Instead, entailment inferences are performed directly over lexical-syntactic representations, as typically obtained from syntactic parsing. Actual meaning judgments are made only by humans evaluating the system.

We define a *language expression* as a syntactically coherent text fragment, having a well-formed fully connected syntactic analysis. *Textual entailment* (entailment, in short) is defined as a relationship between a coherent text T and a language expression, which is considered as a hypothesis, H. We say that T entails H (H is a consequent of T), denoted by $T \Rightarrow H$, if the meaning of H, as interpreted in the context of T, can be inferred from the meaning of T. Motivated by typical application settings, we distinguish between two types of hypotheses: If H is a sentential expression: H is entailed by T if its truth value is defined and is set to TRUE whenever interpreted in the setting described by T. For non-sentential hypotheses: H is entailed by T if its existential meaning is TRUE – its meaning exists in the setting described by T.

The definition of textual entailment captures quite broadly the reasoning about language variability needed by different applications. A QA system has

1. Axiom rule: $P(T_1 \xrightarrow{EP} T_2) = \max_{\{entailment\ patterns\ EP\ matching\ T_1,T_2\}} P(EP \text{ holds for } T_1, T_2)$
2. Reflexivity: $P(T_1 \xrightarrow{REFL} T_2) = 1 \text{ if } T_1 = T_2, \ 0 \text{ otherwise}$
3. Monotone extension: $P(E(T_1) \xrightarrow{MON} E(T_2)) = \mathrm{P}(E \text{ is a } (upward) \text{ monotone extension for } T_1, T_2) \cdot \mathrm{P}(T_1 \Rightarrow T_2)$
4. Restrictive extension: $P(E(T_1) \xrightarrow{RES} T_1) = \mathrm{P}(E \text{ is a restrictive extension for } T_1)$
5. Transitive Chaining: $P(T \Rightarrow H) = \max_m \max_{T_1,T_2,\ldots,T_m} P(T \xrightarrow{*} T_1) \cdot P(T_1 \xrightarrow{*} T_2) \cdot \ldots \cdot P(T_m \xrightarrow{*} H)$

Table 1: Inference Rules (In rule 5: '*' stands for any of the rules 1-4).

1	novel => book	Axiom, by corresponding pattern
2	Bought a novel => bought a book	*MON*, [1] & *E(X)*="bought a *X*"
3	Bought => purchased	Axiom (morphological inflection)
4	Bought a book => purchased a book	*MON*, [3] & E*(X)*="*X* a book"
5	Bought a novel => purchased a book	*TRANS*, [2] & [4]
6	John bought a novel => John purchased a book	*MON*, [5] & E*(X)*="John *X* "
7	John bought a novel yesterday => John bought a novel	*RES*, [6] & E*(X)*="*X* yesterday"
8	John bought a novel yesterday => John purchased a book	*TRANS*, [6]&[7]

Table 2: An example inference chain for "John bought a novel yesterday" =>"John purchased a book"

to identify texts that entail the expected answer. For example, given the question "Who killed Kennedy?", the text "the assassination of Kennedy by Oswald" entails the sentential hypothesis "Oswald killed Kennedy", and therefore constitutes an answer. Similarly, in IR the concepts denoted by a (non-sentential) query expression should be entailed from relevant retrieved documents. In multi document summarization a redundant sentence or expression, to be omitted from the summary, should be entailed from other expressions in the summary. In IE entailment holds between different text variants that express the same relation. And in reference resolution the antecedent typically entails the referring expression (e.g. *IBM* and *company*).

3 The Inference Model

We propose an inference model that approximates the textual entailment relationship, predicting whether an entailment holds for a given text-hypothesis pair. The model consists of a knowledge base of entailment patterns along with a set of inference rules and related probability estimations.

The abstract definition of entailment as presented above is deterministic – for a given text *T* and hypothesis *H* we assume that either $T \Rightarrow H$ holds or not. Our model utilizes a fuzzy notion of entailment by assigning a probability score for an entailment instance, which estimates the probability that entailment indeed holds for this particular text-hypothesis pair.

3.1 Entailment Patterns

We define a *template* as a language expression along with its syntactic analysis, optionally with variables replacing sub-parts of the structure. Variables may be typed syntactically (by the syntactic representation language in use, such as part of speech or relation type in dependency parsing). An entailment pattern consists of:

A. Pattern Structure: an entailing template (left hand side – LHS) and an entailed template (right hand side – RHS), which share the same variable scope.
B. Pattern Probabilities: Prior and contextual (posterior).

E.g.: X $\leftarrow^{subj}$ buy $^{obj}\rightarrow$Y $\Rightarrow$ X $\leftarrow^{subj}$ own $^{obj}\rightarrow$Y.

An entailment pattern specifies that for any instantiation of the variables there is a probability P that a text that entails the LHS entails also the RHS. Probability is estimated as a proper combination of the prior and posterior probabilities of the pattern when applied in the context of a given text.

3.2 Inference Rules

The inference mechanism is designed to use a given repository of entailment patterns and apply a

Hypothesis	Entailing expression	X	Y	Probability
'Oswald$\leftarrow^{subj}$ kill $^{obj}\rightarrow$X'	"Oswald murdered Kennedy"	Kennedy		0.92
	"Lee Oswald, who assassinated Kennedy "	Kennedy		0.85
'X$\leftarrow^{subj}$ buy $^{obj}\rightarrow$Y'	"Yahoo's acquisition of Overture"	Yahoo	Overture	0.78
	"Netscape was bought by AOL"	AOL	Netscape	0.86

Table 3: Example inputs and outputs of the inference engine

probabilistic inference logic compositionally in order to infer entailment between larger expressions. Table 1 lists the core inference rules used by the inference model.

The first rule calculates the maximal probability over all the matching entailment patterns. We call this probability "axiom probability" because entailment patterns are given to the inference engine rather than being deduced. Rules 3&4 describe two ways in which the antecedent and the consequent can be composed into larger expressions while preserving entailment. To represent such compositions, we define first an extension operator for language expressions, denoted $E(T)$, which maps T to a larger language expression in which T is fully embedded (preserving T's syntactic structure).

The Restrictivity rule applies whenever the antecedent could be extended without violating the entailment of the consequent. For example: *French president* $\Rightarrow$ *president*. However, not all extensions preserve expression meaning as *vice president* =/=> *president*. The Monotonicity rule applies whenever an identical extension, which is applied to both the antecedent and the consequent, does not change the validity of the entailment. For example: *Paris* $\Rightarrow$ *France* and hence *visited Paris* $\Rightarrow$ *visited France*. Of course, not all extensions exhibit this kind of monotonicity as *the population of Paris* =/=> *the population of France*. Finally, transitive chaining of rules states that the probability of a complete entailment is the maximal product of probabilities for a chain of rules that derives the hypothesis from the text.

Table 2 demonstrates an example inference chain for: "John bought a novel yesterday" => "John purchased a book" (omitting probabilities).

3.3 The inference Engine

We implement the inference model by a Prolog style engine. The engine operates relative to a given corpus, a knowledge base of entailment patterns and an implementation of the inference rules. It gets a hypothesis as input and applies the inference model to find occurrences of entailing texts in the corpus. For each such text the engine outputs the corresponding variable instantiations, the entailment probability score and a trace of the entailment reasoning. Table 3 presents possible inputs and outputs of the inference engine. The system is evaluated by judging output correctness, measuring precision and recall.

4 Acquisition of Entailment Patterns

The proposed framework of textual entailment provides a generic setting for recognizing language variability. The implementation of such a model poses two challenging areas of learning tasks concerning the acquisition of knowledge needed by the model. Empirical Modeling of monotonicity and restrictivity extensions is, as far as we know, a novel task, and we are now conducting initial studies on how it can be approached (in joint work with Yoad Winter from the Technion, Haifa).The task of learning entailment patterns (structure and probabilities) is related to the problem of automatic paraphrase acquisition, which recently drew noticeable attention of researchers in various application areas. Lin and Pantel (2001) proposed using a distributional similarity approach for extracting "inference rules" for QA. The more dominant approach to paraphrase learning is *instance-based* (sentence-based), which was proposed in the contexts of QA, text generation, summarization, IE and translation (Barzilay and McKeown, 2001; Shinyama et al. 2002; Barzilay and Lee, 2003; Pang et al., 2003; Glickman and Dagan, 2003). The idea is to find pairs (or sets) of matching text fragments that seem to describe roughly the same fact, and share common lexical terms that serve as a set of "anchors". Corresponding components, which share the same relationships with the known anchors, are acquired as paraphrase patterns. E.g., from the fragments "Yahoo bought Overture" and "Yahoo owns Overture" one can deduce the pattern "$X \leftarrow^{subj}$ *buy* $^{obj}\rightarrow Y$" $\Rightarrow$ "$X \leftarrow^{subj}$ own $^{obj}\rightarrow Y$" with "Yahoo" and "Overture" as anchors.

One may view the problem of acquiring entailment patterns as embedding two types of learning tasks: Unsupervised acquisition of candidate patterns and probabilistic binary classification of pattern entailment. In the following subsections we outline a couple of approaches for learning entailment pat-

1-	<fall, rise>	6+	<drop, fall>	62+	<honor, honour>	362+	<bring, take>
2+	<close, end>	7+	<regard, view>	122+	<advance, rise>	422+	<note, say>
3+	<post, report>	8+	<cut, lower>	182+	<benefit, bolster>	482-	<export, load>
4+	<recognize, recognize>	9-	<rise, shed>	242+	<approve, authorize>	542+	<downgrade, relax>
5+	<fire, launch>	10+	<fall, slip>	302+	<kill, slaughter>	602+	<create, establish>

Table 4: Example of verb lexical paraphrases extracted from a subset of the Reuters Corpus (along with annotator's judgments).

tern structure from unlabeled data and empirical estimation of pattern probabilities.

4.1 Extracting paraphrases from a single corpus

Most attempts to paraphrase learning were based on identifying corresponding sentences in parallel or 'comparable' corpora, where each corpus is known to include texts that largely correspond to texts in another corpus (Barzilay and McKeown 2001, Shinyama et al. 2002, Pang et al. 2003, Barzilay and Lee 2003). The major types of comparable corpora are different translations of the same text, and multiple news sources that overlap largely in the stories that they cover. In (Glickman and Dagan, 2003) we proposed an instance-based algorithm for acquiring lexical paraphrases from a single corpus. Clearly, requiring a pair (or set) of comparable corpora is a disadvantage, since such corpora do not exist for all domains, and are substantially harder to assemble. We therefore developed a method that detects concrete paraphrase instances within a single corpus. Such paraphrase instances can be found since a coherent domain corpus is likely to include repeated references to the same concrete facts or events, even though they might be found within generally different stories. The method combines statistical and linguistic filters to produce a probabilistically motivated paraphrase likelihood score. We compared our method to the vector-based approach of (Lin and Pantel 2001). Our instance-based approach seems to help assessing the reliability of candidate paraphrases, which is more difficult to assess by global distributional similarity measures such as the measure of Lin and Pantel.

Table 4, shows top paraphrases extracted from a subset of the Reuters RCV1 Corpus along with annotator's judges.

4.2 Learning entailment patterns from the web

We are currently following the instance-based paradigm for unsupervised learning of paraphrase patterns from plain corpora or the Web, extending it to obtain broader coverage and to fit the structure of our entailment-based framework. The learning process consists of two main tasks: (1) identifying reliable sets of anchors and (2) identifying the various templates that connect the anchors and take part in the entailment patterns. Inspired by earlier work on learning variations of pre-specified relations from the web (Agichtein and Gravano, 2000; Ravichandran and Hovy 2002; Duclaye et al., 2002), we propose using a bootstrapping approach that performs the two tasks iteratively, following the general co-training scheme (Blum and Mitchell, 1998; Collins and Singer, 1999; Abney, 2002). The special challenge in this task is to search for good entailment patterns (paraphrases) involving *any* given lexical item, without relying on specific anchors that were identified before hand and therefore dictate the identity of the entailment patterns which may be identified. This work is carried in a joint project with Hristo Tanev and Bonaventura Coppola from ITC-IRST (at Trento) and Idan Szpektor from Tel Aviv University.

Core	Example Anchor Sets extracted for the core, and example sentences containing the anchor set	Extracted Candidate Templates for entailment
acquire	{Novell(X), SuSE Linux(Y)}	X's purchase of Y
	"Novell's purchase of SuSE Linux earlier this month has sparked varied speculation"	
	{Niagara Mohawk(X), National Grid(Y)}	Y will become owned subsidiary of X
	"Upon completion of the transaction, Niagara Mohawk will become a wholly-owned subsidiary of National Grid"	
approve	{shareholders(X), company's merger(Y)}	X voted in favor of Y
	"Shareholders of Three Rivers Bancorp Inc. overwhelmingly voted in favor of the company's merger with Sky Financial Group Inc."	
	{shareholders(X), transaction(Y)}	
	"… shareholders of both companies have overwhelmingly voted in favor of the merger transaction."	
kill	{explosion(X), 21 people(Y)}	Y dies in X
	"21 people died in an explosion in the capital of Sri Lanka"	

Table 5: Examples of algorithm output

The first phase identifies reliable anchor sets for a given lexical "core" word or term, for which we want to find paraphrases (entailment patterns). An anchor set is a set of terms which indicates with a high probability that a common fact is described in multiple sentences. Iterative web search queries are performed to retrieve sentences containing the core term and associated anchors. Various statistical criteria are then applied over the retrieved anchor candidates to identify "promising" anchor sets. For example, given the core term 'murder' the following are among the resulting anchor sets: <Kennedy, Oswald>, <Nicole Brown, O.J. Simpson>.

For the second phase we are using an algorithm developed at ITC-IRST in the framework of the project MoreWeb founded by the Province of Trento. The algorithm identifies the most general (smallest) linguistic structures across multiple anchor sets that connect anchors in the parsed sentences. The templates are achieved by replacing the anchors with variables in the structures. This task resembles ILP-style symbolic learning.

The bootstrapping scheme consists of iterating between these two procedures, as summarized in Table 6. Table 5 shows examples of initial outputs that were obtained from one iteration of the two phases (the table presents some of the more complex extracted templates; in addition, many simpler templates that correspond to synonyms of the core were extracted, similar to those extracted from the Reuters corpus, as demonstrated in Table 4).

I) Initialization/Seeding: Initialize the list of candidates for template cores from an input lexicon (extracted from a dictionary, WordNet, domain corpus etc.).

II) For each template core candidate:
   a. Extract sentences containing the template core terms using querying tools
   b. Extract candidate anchor sets from these sentences, testing statistical significance

III) For each extracted anchor set:
   a. Extract a set of sentences containing the anchor set terms
   b. Extract candidate templates and cores from matching sub-structures, testing significance

IV) Iterate II & III (until acquisition saturates)

V) Generate entailment patterns between the extracted templates and estimate their probabilities (Section 4.3).

Table 6: bootstrapping algorithm outline

4.3 Learning Pattern probabilities

	{Kennedy, Oswald}	{Lincoln, Booth}	{Nicole Brown, O.J. Simpson}
assassinate	2478	2692	0
murder	604	271	251

Table 7: web search counts (from www.av.com) for various cores (rows) and anchor sets (columns).

The iterative template extraction process produces a contingency table of frequency counts for anchor sets and templates (demonstrated in Table 7). Entailment patterns and their estimated entailment probabilities will be derived from the resulting contingency table. Table 7 demonstrates that "assassinate" entails murder with high probability but

entailment in the other direction holds only in part of the cases. We plan to investigate appropriate estimation of prior entailment probabilities from the table rows. Estimating pattern probability is challenging since the data is not labeled for entailment.

The contexts in which templates occur determine a posterior contextual probability for the applicability of the corresponding patterns. For example, to learn (from Table 7) the contexts in which "murder" entails "assassinate" one needs to identify typical contexts for the two left columns, corresponding to political settings, which distinguish them from the non-political setting of the right column. We notice that this task resembles the Word Sense Disambiguation (WSD) classification task. In fact, a major motivation for using contextual probabilities is to apply correctly patterns involving ambiguous words (E.g., "bank⇒company" only in its financial sense). Accordingly, we are exploring ways to utilize WSD representation and learning schemes to learn contextual probabilities. Finally, another challenging task is to combine properly the prior and posterior estimates when applying a pattern, based on the degree of context match.

5 Conclusion

Textual entailment plays an important role within natural language applications. We propose a generic model for capturing such textual entailment and describe our first steps in its implementation. The general approach of utilizing a "shallow" level of semantics is not novel - many systems perform inferences based on lexical-syntactic structures and knowledge sources. However this is usually done individually in each system in a limited and often ad-hoc manner. The proposed framework supplies a principled mechanism for combining in a single inference multiple pieces of knowledge, which stem from various knowledge sources. A primary research goal is to find out 'how far' one can get by performing such inference directly over lexical-syntactic representations, while avoiding semantic inference over explicit meaning-level representations.

Acknowledgements

We would like to highlight the participation of additional researchers in the research directions described in this paper. Learning of paraphrase patterns from the Web is carried in a joint project with Hristo Tanev and Bonaventura Coppola from ITC-IRST (Trento) and Idan Szpektor from Tel Aviv University. The study of compositional combination of entailment patterns is done jointly with Yoad Winter from the Technion, Haifa. We would further like to thank Hans Uszkoreit, Moshe Koppel and Dan Roth, for fruitful and encouraging discussions related to the probabilistic entailment framework. The work of the first author in these projects is partly funded by ITC-IRST.

References

Steven Abney. 2002. *Bootstrapping*. ACL.

Eugene Agichtein and Luis Gravano. 2000. *Snowball: Extracting relations from large plain-text collections*. In Proceedings of the 5th ACM International Conference on Digital Libraries (DL'00).

Regina Barzilay, Lillian Lee. 2003. *Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment*. NAACL-HLT.

Regina Barzilay, Kathleen McKeown. 2001. *Extracting Paraphrases from a Parallel Corpus*. ACL/EACL.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. *CONLT*, 1998.

Michael Collins and Yoram Singer. 1999. *Unsupervised models for named entity classification*. Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.

Oren Glickman, Ido Dagan. 2003. *Identifying Lexical Paraphrases From a Single Corpus: A Case Study for Verbs*. Proceedings of Recent Advantages in Natural Language Processing (RANLP '03).

Dekang Lin and Patrick Pantel. 2001. *Discovery of Inference Rules for Question Answering*. Natural Language Engineering 7(4):343-360.

Florence Duclaye, François Yvon and Olivier Collin. 2002. *Using the Web as a Linguistic Resource for Learning Reformulations Automatically*. LREC.

Yusuke Shinyama, Satoshi Sekine, Kiyoshi Sudo and Ralph Grishman. 2002. *Automatic Paraphrase Acquisition from News Articles*. Proceedings of the Human Language Technology Conference (HLT).

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. *Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences*. *HLT/NAACL*.

Ravichandran, D. and E.H. Hovy. 2002. *Learning Surface Text Patterns for a Question Answering System*. In Proceedings of the 40th ACL conference.