A Probabilistic Lexical Model for Ranking Textual Inferences

Eyal Shnarch and **Ido Dagan**
Computer Science Department
Bar-Ilan University
Ramat-Gan 52900, Israel
{shey,dagan}@cs.biu.ac.il

Jacob Goldberger
Faculty of Engineering
Bar-Ilan University
Ramat-Gan 52900, Israel
goldbej@eng.biu.ac.il

Abstract

Identifying textual inferences, where the meaning of one text follows from another, is a general underlying task within many natural language applications. Commonly, it is approached either by generative syntactic-based methods or by "lightweight" heuristic lexical models. We suggest a model which is confined to simple lexical information, but is formulated as a principled generative probabilistic model. We focus our attention on the task of *ranking textual inferences* and show substantially improved results on a recently investigated question answering data set.

1 Introduction

The task of identifying texts which share semantic content arises as a general need in many natural language processing applications. For instance, a paraphrasing application has to recognize texts which convey roughly the same content, and a summarization application needs to single out texts which contain the content stated by other texts. We refer to this general task as *textual inference* similar to prior use of this term (Raina et al., 2005; Schoenmackers et al., 2008; Haghighi et al., 2005).

In many textual inference scenarios the setting requires a classification decision of whether the inference relation holds or not. But in other scenarios ranking according to inference likelihood would be the natural task. In this work we focus on *ranking textual inferences*; given a sentence and a corpus, the task is to rank the corpus passages by their plausibility to imply as much of the sentence meaning as possible. Most naturally, this is the case in question answering (QA), where systems search for passages that cover the semantic components of the question. A recent line of research was dedicated to this task (Wang et al., 2007; Heilman and Smith, 2010; Wang and Manning, 2010).

A related scenario is the task of Recognizing Textual Entailment (RTE) within a corpus (Bentivogli et al., 2010)[1]. In this task, inference systems should identify, for a given *hypothesis*, the sentences which entail it in a given corpus. Even though RTE was presented as a classification task, it has an appealing potential as a ranking task as well. For instance, one may want to find texts that validate a claim such as *cellular radiation is dangerous for children*, or to learn more about it from a newswire corpus. To that end, one should look for additional mentions of this claim such as *extensive usage of cell phones may be harmful for youngsters*. This can be done by ranking the corpus passages by their likelihood to entail the claim, where the top ranked passages are likely to contain additional relevant information.

Two main approaches have been used to address textual inference (for either ranking or classification). One is based on transformations over syntactic parse trees (Echihabi and Marcu, 2003; Heilman and Smith, 2010). Some works in this line describe a probabilistic generative process in which the parse tree of the question is generated from the passage (Wang et al., 2007; Wang and Manning, 2010).

In the second approach, lexical models have been employed for textual inference (MacKinlay and Baldwin, 2009; Clark and Harrison, 2010). Typi-

[1] http://www.nist.gov/tac/2010/RTE/index.html

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 237–245,
Montréal, Canada, June 7-8, 2012.

cally, lexical models consider a text fragment as a bag of terms and split the inference decision into two steps. The first is a *term-level* estimation of the inference likelihood for each term independently, based on direct lexical match and on lexical knowledge resources. Some commonly used resources are WordNet (Fellbaum, 1998), distributional-similarity thesauri (Lin, 1998), and web knowledge resources such as (Suchanek et al., 2007). The second step is making a final *sentence-level* decision based on these estimations for the component terms. Lexical models have the advantage of being fast and easy to utilize (e.g. no dependency on parsing tools) while being highly competitive with top performing systems, e.g. the system of Majumdar and Bhattacharyya (2010).

In this work, we investigate how well such lexical models can perform in textual inference ranking scenarios. However, while lexical models usually apply heuristic methods, we would like to pursue a principled learning-based generative framework, in analogy to the approaches for syntactic-based inference. An attractive work in this spirit is presented in (Shnarch et al., 2011a), that propose a model which is both lexical and probabilistic. Later, Shnarch et al. (2011b) improved this model and reported results that outperformed previous lexical models and were on par with state-of-the-art RTE models.

Whereas their term-level model provides means to integrate lexical knowledge in a probabilistic manner, their sentence-level model depends to a great extent on heuristic normalizations which were introduced to incorporate prominent aspects of the sentence-level decision. This deviates their model from a pure probabilistic methodology.

Our work aims at amending this deficiency and proposes a new probabilistic sentence-level model based on a Markovian process. In that model, all parameters are estimated by an EM algorithm. We evaluate this model on the tasks of ranking passages for QA and ranking textual entailments within a corpus, and show that eliminating the need for heuristic normalizations greatly improves state-of-the-art performance. The full implementation of our model is available for download[2] and can be used as an easy-to-install and highly competitive inference engine that operates only on lexical knowledge, or as a lexical component integrated within a more complex inference system.

2 Background

Wang et al. (2007) provided an annotated data set, based on the Text REtrieval Conference (TREC) QA tracks[3], specifically for the task of ranking candidate answer passages. We adopt their experimental setup and next review the line of syntactic-based works which reported results on this data set.

2.1 Syntactic generative models

Wang et al. (2007) propose a quasi-synchronous grammar formulation which specifies the generation of the question parse tree, loosely conditioned on the parse tree of the candidate answer passage. Their model showed improvement over previous syntactic models for QA: Punyakanok et al. (2004), who computed similarity between question-answer pairs with a generalized tree-edit distance, and Cui et al. (2005), who developed an information measure for sentence similarity based on dependency paths of aligned words. Wang et al. (2007) reproduced these methods and extended them to utilize WordNet.

More recently, Heilman and Smith (2010) improved Wang et al. (2007) results with a classification based approach. Feature for the classifier were extracted from a greedy algorithm which searches for tree-edit sequences which transform the parse tree of the candidate answer into the one of the question. Unlike other works reviewed here, this one does not utilize lexical knowledge resources.

Similarly, Wang and Manning (2010) present an extended tree-edit operations set and search for edit sequences to generate the question from the answer candidate. Their CRF-based classifier models these sequences as latent variables.

An important merit of these methods is that they offer principled, often probabilistic, generative models for the task of ranking candidate answers. Their drawback is the need for syntactic analysis which makes them slower to run, dependent on parsing performance, which is often mediocre in many text genres, and inadequate for languages which lack proper parsing tools.

[2] http://www.cs.biu.ac.il/~nlp/downloads/probLexModel.html

[3] http://trec.nist.gov/data/qamain.html

2.2 Lexical models

Lexical models, on the other hand, are faster, easier to implement and are more practical for various genres and languages. Such models derive from knowledge resources *lexical inference rules* which indicate that the meaning of a lexical term can be inferred from the meaning of another term (e.g. *youngsters* → *children* and *harmful* → *dangerous*). They are common in the Recognizing Textual Entailment (RTE) systems and we present some representative methods for that task. We adopt textual entailment terminology and henceforth use *Hypothesis* (denoted H) for the inferred text fragment and *Text* (denoted T) for the text from which it is being inferred[4].

Majumdar and Bhattacharyya (2010) utilized a simple union of lexical rules derived from various lexical resources for the term-level step. They derived their sentence-level decision based on the number of matched hypothesis terms. The results of this simple model were only slightly worse than the best results of the RTE-6 challenge which were achieved by a syntactic-based system (Jia et al., 2010). Clark and Harrison (2010), on the other hand, considered the number of mismatched terms in establishing their sentence-level decision. MacKinlay and Baldwin (2009) represented text and hypothesis as word vectors augmented with lexical knowledge. For sentence-level similarity they used a variant of the cosine similarity score. Common to most of these lexical models is the application of heuristic methods in both the term and the sentence level steps.

Targeted to replace heuristic methods with principled ones, Shnarch et al. (2011a) present a model which aims at combining the advantages of a probabilistic generative model with the simplicity of lexical methods. In some analogy to generative parse-tree based models, they propose a generative process for the creation of the hypothesis from the text.

At the term-level, their model combines knowledge from various input resources and has the advantages of considering the effect of transitive rule application (e.g. *mobile phone* → *cell phone* → *cellular*) as well as the integration of multiple pieces of evidence for the inference of a term (e.g. both the appearance of *harmful* and *risky* in T provide evidence for the inference of *dangerous* in H). We denote this term-level Probabilistic Lexical Model as PLM^{TL}, and have reproduced it in our work as presented in Section 4.1. For the sentence-level decision they describe an AND gate mechanism, i.e. deducing a positive inference decision for H as a whole only if all its terms were inferred from T.

In an extension to that work, Shnarch et al. (2011b) modified PLM^{TL} to improve the sentence-level step. They pointed out some prominent aspects for the sentence-level decision. First, they suggest that a hypothesis as a whole can be inferred from the text even if some of its terms are not inferred. To model this, they introduced a noisy-AND mechanism (Pearl, 1988). Additionally, they emphasized the effect of hypothesis length and the dependency between terms on the sentence-level decision. However, they did not fully achieve their target of presenting a fully coherent probabilistic model, as their model included heuristic normalization formulae.

On the contrary, the model we present is the first along this line to be fully specified in terms of a generative setting and formulated in pure probabilistic terms. We introduce a Markovian-style probabilistic model for the sentence-level decision. This model receives as input term-level probabilistic estimates, which may be provided by any term-level model. In our implementation we embed PLM^{TL} as the term-level model and present a complete coherent Markovian-based Probabilistic Lexical Model, which we term *M-PLM*.

3 Markovian sentence-level model

The goal of a sentence-level model is to integrate term-level inputs into an inference decision for the hypothesis as a whole. For a hypothesis $H = h_1, \ldots, h_n$ and a text T, term-level models first estimate independently for each term h_t its probability to be inferred from T. Let x_t be a binary random variable representing the event that h_t is indeed inferred from T (i.e., $x_t = 1$ if h_t is inferred and 0 otherwise).

Given these term-level probabilities, a sentence-level model is employed to estimate the probability that H as a whole is inferred from T. This step is

[4] In the task of passage ranking for QA, the hypothesis is the question and the text is the candidate passage.

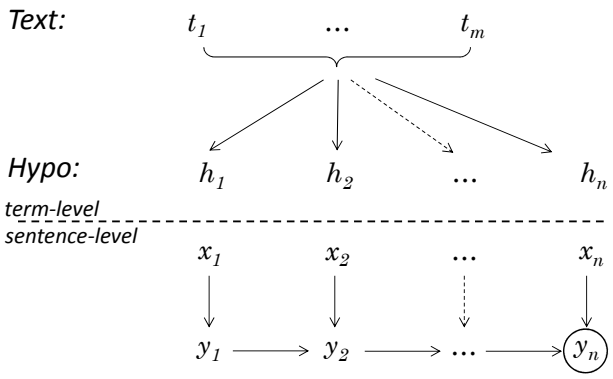

Figure 1: A probabilistic lexical model: the upper part is the term-level input to the sentence-level Markovian process, depicted in the lower part. x_i is a binary variable representing the inference of h_i and y_j is a variable for the accumulative inference decision for the first j terms of *Hypo*. The final sentence-level decision is given by y_n.

the focus of our work. We assume that the term-level probabilities are given as input. Section 4.1 describes *PLM*^{TL}, as a concrete method for deriving these probabilities.

Our sentence-level model is based on a Markovian process and is described in Section 3.1. In particular, it takes into account, in probabilistic terms, the prominent factors in lexical entailment, mentioned in Section 2. An efficient inference algorithm for our model is given in Section 3.2 and EM-based learning is specified in Section 3.3.

3.1 Markovian sentence-level decision

The motivation for proposing a Markovian process for the sentence-level is to establish an intermediate model, lying between two extremes: assuming full independence between hypothesis terms versus assuming that every term is dependent on all other terms. The former alternative is too weak, while the latter alternative is computationally hard and not very informative, and thus hard to capture in a model. Our model specifies a Markovian dependence structure, which limits the dependence scope to adjacent terms, as follows.

We define a binary variable y_t to be the accumulated sentence-level inference decision up to h_t. In other words, $y_t=1$ if the subset $\{h_1,\ldots,h_t\}$ of H's terms is inferred as a whole from T.

Note that this means that y_t can be 1 even if some terms amongst $h_1,\ldots,h_t$ are not inferred. As y_n is the decision for the complete hypothesis, our model addresses this way the prominent aspect that the hypothesis as a whole may be inferred even if some of its terms are not inferred. The reason for allowing this is that such un-inferred terms may be inferred from the global context of T, or alternatively, are actually inferred from T but the knowledge resources in use do not contain the proper lexical rule to make such inference.

Figure 1 describes both steps of a full lexical inference model. Its lower part depicts our Markovian process. In the proposed model the inference decision at each position t is a combination of x_t, the variable for the event of h_t being inferred, and y_{t-1}, the accumulated decision at the previous position. Therefore, the *transition* parameters of *M-PLM* can be modeled as:

$$q_{ij}(k)=P(y_t=k|y_{t-1}=i,x_t=j)\quad \forall k,i,j\in\{0,1\}$$

where $y_1=x_1$. For instance, $q_{01}(1)$ is the probability that $y_t=1$, given that $y_{t-1}=0$ and $x_t=1$.

Applying the Markovian process on the entire hypothesis we get y_n, which represents the final sentence-level decision, where a soft decision is obtained by computing the probability of $y_n=1$:

$$P(y_n=1)=\sum_{\substack{x_1,\ldots,x_n\\ y_2,\ldots,y_{n-1},y_n=1}} P(x_1)\prod_{t=2}^{n}P(x_t)P(y_t|y_{t-1},x_t)$$

The summation is done over all possible binary values of the term-level variables $x_1,\ldots,x_n$ and the accumulated sentence-level variables $y_2,\ldots,y_{n-1}$ where $y_n=1$. Note that for clarity, in this formula x_t and y_t denote the binary *values* at the corresponding variable positions. A tractable form for computing $P(y_n=1)$ is presented in Section 3.2.

Overall, the prominent factors in lexical entailment, raised by prior works, are incorporated within the core structure of this probabilistic model, without the need to resort to heuristic normalizations. Reducing the negative affect of hypothesis length on the entailment probability is achieved by having y_t, at each position, being *directly* dependent only on x_t and y_{t-1} as opposed to being affected by all hypothesis terms. The second factor, modeling the dependency between hypothesis terms, is addressed by the

indirect dependency of y_n on all preceding hypothesis terms. This dependency arises from the recursive nature of the Markovian model, as can be seen in the next section.

Our proposed Markovian process presents a linear dependency between terms which, to some extent, poses an anomaly with respect to the structure of the entailment phenomenon. Yet, as we do want to limit the dependence structure, following the natural order of the sentence words seems the most reasonable choice, as common in many other types of sequential models. We also tried randomizing the word order which, on average, did not improve performance.

3.2 Inference

The accumulated sentence-level inference can be efficiently computed using a typical forward algorithm. We denote the probability of $x_t = j, j \in \{0,1\}$ by $h_t(j) = P(x_t = j)$. The forward step is given in Eq. (1) and its initialization is defined in Eq. (2).

$$\alpha_t(k) = P(y_t = k) = \sum_{i,j \in \{0,1\}} \alpha_{t-1}(i) h_t(j) q_{ij}(k) \quad (1)$$

$$\alpha_1(k) = P(x_1 = k) \quad (2)$$

where $k \in \{0,1\}$ and $t = 2, ..., n$.

$\alpha_t(k)$ is the probability that the accumulated decision at position t is k. It is calculated by summing over the probabilities of all four combinations of $\alpha_{t-1}(i)$ and $h_t(j)$, multiplied by the corresponding transition probability, $q_{ij}(k)$.

The soft sentence-level decision can be efficiently calculated by:

$$P(y_n = a) = \alpha_n(a) \qquad a \in \{0,1\} \quad (3)$$

3.3 Learning

Typically, natural language applications work at the sentence-level. The training data for such applications is, therefore, available as annotations at the sentence-level. Term-level alignments between passage terms and question terms are rarely available. Hence, we learn our term-level parameters from available sentence-level annotations, using the generative process described above to bridge the gap between these two levels.

For learning we use the typical backwards algorithm which is described by Eq. (4) and Eq. (5), where $\beta_t(a|i)$ is the probability that the full hypothesis inference value is a given that $y_t = i$.

$$\beta_n(a|i) = P(y_n = a | y_n = i) = 1_{\{a=i\}} \quad (4)$$

$$\beta_t(a|i) = P(y_n = a | y_t = i) = \sum_{j,k \in \{0,1\}} h_{t+1}(j) q_{ij}(k) \beta_{t+1}(a|k) \quad (5)$$

where $t = n-1, .., 1$, $a \in \{0,1\}$ and $1_{\{condition\}}$ is the indicator function which returns 1 if $condition$ holds and 0 otherwise.

To estimate $q_{ij}(k)$, the parameters of the Markovian process, we employ the EM algorithm:

E-step: For each (T, H) pair in the training data set, annotated with $a \in \{0,1\}$ as its sentence-level inference value, we evaluate the expected probability of every transition given the annotation value a:

$$w_{tijk}(T,H) = P(y_{t-1} = i, x_t = j, y_t = k | y_n = a) = \frac{\alpha_{t-1}(i) h_t(j) q_{ij}(k) \beta_t(a|k)}{P(y_n = a)} \quad (6)$$

$\forall i, j, k \in \{0,1\}$ and $t = 2, ..., |H|$.

M-step: Given the values of $w_{tijk}(T,H)$ we can estimate each $q_{ij}(1)$, $i, j \in \{0,1\}$, by taking the proportion of transitions in which $y_{t-1} = i$, $x_t = j$ and $y_t = 1$, out of the total transitions in which $y_{t-1} = i$ and $x_t = j$:

$$q_{ij}(1) \leftarrow \frac{\sum_{(T,H)} \sum_{t=2}^{|H|} w_{tij1}(T,H)}{\sum_{(T,H)} \sum_{t=2}^{|H|} \sum_{k \in \{0,1\}} w_{tijk}(T,H)} \quad (7)$$

$q_{ij}(0) = 1 - q_{ij}(1)$

4 Complete model implementation

We next describe the end-to-end probabilistic lexical inference model we used in our evaluations. We implemented *PLM*^{TL} as our term-level model to provide us with $h_t(j)$, the term-level probabilities. We chose this model since it is fully lexical, has the advantages of lexical knowledge integration described in Section 2 and achieved top results on RTE data sets. Next, we summarize *PLM*^{TL}, and in Appendix A we show how to adjust the learning schema to fit into our sentence-level model.

4.1 *PLM*^{TL}

Shnarch et al. (2011a) provide a term-level model which integrates lexical rules from various knowledge resources. As described below it also considers transitive chains of rule applications as well as the impact of parallel chains which provide multiple evidence that $h \in H$ is inferred from T.

Their model assumes a parameter θ_R for each knowledge resource R in use. θ_R specifies the resource's reliability, i.e. the prior probability that applying a rule from R to an arbitrary text-hypothesis pair would yield a valid inference.

Next, transitive *chains* may connect a text term to a hypothesis term via intermediate term(s). For instance, starting from the text term *T-Mobile*, a chain that utilizes the lexical rules *T-Mobile* $\rightarrow$ *telecom* and *telecom* $\rightarrow$ *cell phone* enables the inference of the term *cell phone* from T. They compute, for each step in a chain, the probability that this step is valid based on the θ_R values. Denoting the resource which provided a rule r by $R(r)$, Eq. (8) specifies that the validity probability of the inference step corresponding to the application of the rule r within the chain c pointing at h_t (as represented by x_{tcr}) is $\theta_{R(r)}$.

Next, for a chain c pointing at h_t (represented by x_{tc}) to be valid, *all* its rule steps should be valid for this pair. Eq. (9) estimates this probability by the joint probability that the applications of all rules $r \in c$ are valid, assuming independence of rules.

Several chains may connect terms in T to h_t, thus providing multiple pieces of evidence that h_t is inferred from T. For instance, both *youngsters* and *kids* in T may indicate the inference of *children* in H. For a term h_t to be inferred from the entire sentence T it is enough that *at least one* of the chains from T to h_t is valid. This is the complement event of h_t not being inferred from T which happens when all chains which suggest the inference of h_t, denoted by $C(h_t)$, are invalid. Eq. (10) specifies this probability (again assuming independence of chains).

$$P(x_{tcr}=1) = \theta_{R(r)} \quad (8)$$

$$P(x_{tc}=1) = \prod_{r \in c} P(x_{tcr}=1) \quad (9)$$

$$h_t(1) = P(x_t=1) = 1 - P(x_t=0) = 1 - \prod_{c \in C(h_t)} P(x_{tc}=0) \quad (10)$$

With respect to the contributions of our work, we note that previous works resorted to applying some heuristic amendments on these equations to achieve valuable results. In contrast, our work is the first to present a purely generative model. This achievement shows that it is possible to shift from ad-hoc heuristic methods, which are common practice, to more solid mathematically-based methods.

Finally, for ranking text passages from a corpus for a given hypothesis (question in the QA scenario), our Markovian sentence-level model takes as its input the outcome of Eq. (10) for each $h_t \in H$. For *PLM*^{TL} we need to estimate the model parameters, that is the various θ_R values. In our Markovian model this is done by the scheme detailed in Appendix A. Given these term-level probabilities, our model computes for each hypothesis its probability to be inferred from each of the corpus passages, namely $P(y_n = 1)$ in Eq (3). Passages are then ranked according to this probability.

5 Evaluations and Results

To evaluate the performance of *M-PLM* for ranking textual inferences we focused on the task of ranking candidate answer passages for question answering (QA) as presented in Section 5.1. Additionally, we demonstrate the added value of our sentence-level model in another ranking experiment based on RTE data sets, described in Section 5.2.

5.1 Answer ranking for question answering

Data set We adopted the experimental setup of Wang et al. (2007) who also provided an annotated data set for answer passage ranking in QA[5].

In their data set an instance is a pair of a factoid question and a candidate answer passage (a single sentence in this data set). It was constructed from the data of the QA tracks at TREC 8–13. The question-candidate pairs were manually judged and a pair was annotated as positive if the candidate passage indicates the correct answer for the question. The training and test sets roughly contain 5700 and 1500 pairs correspondingly.

[5]The data set was kindly provided to us by Mengqiu Wang and is available for download at http://www.cs.stanford.edu/~mengqiu/data/qg-emnlp07-data.tgz.

Method *PLM*^{TL} utilizes WordNet and the Catvar (Categorial Variation) derivations database (Habash and Dorr, 2003) as generic and publicly available lexical knowledge resources, when question and answer terms are restricted to the first WordNet sense. In order to be consistent with (Shnarch et al., 2011b), the best performing model of prior work, we restricted our model to utilize only these two resources which they used. However, additional lexical resources can be provided as input to our model (e.g. a distributional similarity-base thesaurus).

We report Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR), the standard measures for ranked lists. In the cases of tie we took a conservative approach and ranked positive annotated instances below the negative instances scored with the same probability. Hence, the reported figures are lower-bounds for any tie-breaking method that could have been applied.

Results We compared our model to all 5 models evaluated for this data set, described in Section 2, and to our own implementation of (Shnarch et al., 2011b). We term this model Heuristically-Normalized Probabilistic Lexical Model, *HN-PLM*, since it modifies *PLM*^{TL} by introducing heuristic normalization formulae. As explained earlier, both *M-PLM* and *HN-PLM* embed *PLM*^{TL} in their implementation but they differ in their sentence-level model. In our implementation of both models, *PLM*^{TL} applies chains of transitive rule applications whose maximal length is 3.

As seen in Table 1, *M-PLM* outperforms all prior models by a large margin. A comparison of *M-PLM* and *HN-PLM* reveals the major positive effect of choosing the Markovian process for the sentence-level decision. By avoiding heuristically-normalized formulae and having all our parameters being part of the Markovian model, we managed to increases both MAP and MRR by nearly 2.5%[6].

Ablation Test As an additional examination of the impact of the Markovian process components, we evaluated the contribution of having 4 transition parameters. The AND-logic applied by (Shnarch et al., 2011a) to their sentence-level decision roughly corresponds to 2 of the Markovian parameters. A binary AND outputs 1 if both its inputs are 1. This corresponds to $q_{11}(1)$ which is indeed estimate to be near 1. In any other case an AND gate outputs 0. This corresponds to $q_{00}(1)$ which was estimated to be near zero.

The two parameters q_{01} and q_{10} are novel to the Markovian process and do not have counterparts in (Shnarch et al., 2011a). These parameters are the cases in which the sentence-level decision accumulated so far and the term-level decision do not agree. Introducing these 2 parameters enables our model to provide a positive decision for the hypothesis as a whole (or for a part of it) even if some of its terms were not inferred. We performed an ablation test on each of these two parameters by forcing the value of the ablated parameter to be zero. The notable performance drop presented in Table 2 indicates the crucial contribution of these parameters to our model.

System	MAP	MRR
Punyakanok et al.	41.89	49.39
Cui et al.	43.50	55.69
Wang & Manning	59.51	69.51
Wang et al.	60.29	68.52
Heilman & Smith	60.91	69.17
Shnarch et al. *HN-PLM*	61.89	70.24
M-PLM	**64.38**	**72.69**

Table 1: Results (in %) for the task of answer ranking for question answering (sorted by MAP).

Ablated parameter	Δ MAP	Δ MRR
$q_{01}(1) = 0$	-2.61	-4.91
$q_{10}(1) = 0$	-2.12	-2.86

Table 2: Ablation test for the novel parameters of the Markovian process. Results (in %) indicate performance drop when forcing a parameter to be zero.

5.2 RTE evaluations

To assess the added value of our model on an additional ranking evaluation, we utilize the search task data sets of the recent Recognizing Textual Entailment (RTE) benchmarks (Bentivogli et al., 2009; Bentivogli et al., 2010), which were originally con-

[6]The difference is not significant according to the Wilcoxon test, however we note that given the data set size it is hard to get a significant difference and that both Heilman and Smith (2010) and Wang and Manning (2010) improvements over the results of Wang et al. (2007) were not statistically significant.

structed for the task of entailment classification. In that task a hypothesis is given with a corpus and the goal is to identify which sentences of the corpus entail the hypothesis. This setting naturally lends itself to a ranking scenario, in which the desired output is a list of the corpus sentences ranked by their probability to entail the given hypothesis.

To that end, we employed the same methodology as described in the previous section. Table 3 presents the improvement of our model over *HN-PLM*, whose classification performance was reported to be on par with best-performing systems on these data sets[7]. As can be seen, the improvement is substantial for both measures on both data sets. These results further assess the contribution of our Markovian sentence-level model.

	RTE-5		RTE-6	
	MAP	MRR	MAP	MRR
HN-PLM	58.0	82.9	54.0	71.9
M-PLM	61.6	84.8	60.0	79.2
Δ	+3.6	+1.9	+6.0	+7.3

Table 3: Improvements of our sentence-level model over *HN-PLM*. Results (in %) are shown for the last RTE and for the search task in RTE-5.

6 Discussion

This paper investigated probabilistic lexical models for ranking textual inferences focusing on passage ranking for QA. We showed that our coherent probabilistic model, whose sentence-level model is based on a Markovian process, considerably improves five prior syntactic-based models as well as a heuristically-normalized lexical model. Therefore, it raises the baseline for future methods.

In future work we would like to further explore a broader range of related probabilistic models. Especially, as our Markovian process is dependent on term order, it would be interesting to investigate models which are not order dependent.

Initial experiments on the classification task show that *M-PLM* performs well above the average system but below *HN-PLM*, since it does not normalize the estimated probability well across hypothesis. We therefore suggest a future work on better classification models.

Finally, we view this work as joining a line of research which develops principled probabilistic models for the task of textual inference and demonstrates their superiority over heuristic methods.

[7]RTE data sets were only used for the classification task so far, therefore there are no state-of-the-art results to compare with, when utilizing them for the ranking task.

A Appendix: Adaptation of *PLM*^{TL} learning

M-PLM embeds *PLM*^{TL} as its term-level model. *PLM*^{TL} introduces θ_R values as additional parameters for the complete model. We show how we modify (Shnarch et al., 2011a) E-step formula to fit our Markovian modeling, described in Section 3.1. The M-step formula remains exactly the same.

Eq. (11) estimates the a-posteriori validity probability of a single application of the rule r in the transitive chain c pointing at h_t, given that the annotation of the pair is a.

$$w_{tcr}(T,H) = P(x_{tcr}=1|y_n=a) = \frac{\sum_{i,j,k\in\{0,1\}} \alpha_{t-1}(i)P(x_t=j|x_{tcr}=1)\theta_{R(r)}q_{ij}(k)\beta_t(a|k)}{P(y_n=a)} \quad (11)$$

where $t=2\ldots n$ and $P(x_t=j|x_{tcr}=1)$ is the probability that the inference value of x_t is j, given that the application of r provides a valid inference step. As appeared in (Shnarch et al., 2011b) this probability can be evaluated as follows:

$$P(x_t=1|x_{tcr}=1)=1-\frac{P(x_t=0)}{P(x_{tc}=0)}\left(1-\frac{P(x_{tc}=1)}{\theta_{R(r)}}\right)$$

For $t=1$ there is no accumulated sentence-level decision at the previous position (i.e. no α_{t-1}) therefore Eq. (11) becomes:

$$w_{1cr}(T,H) = \frac{\sum_{j\in\{0,1\}}P(x_1=j|x_{1cr}=1)\theta_{R(r)}\beta_1(a|j)}{P(y_n=a)}$$

Acknowledgments

This work was partially supported by the Israel Science Foundation grant 1112/08, the PASCAL-2 Network of Excellence of the European Community FP7-ICT-2007-1-216886, and the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287923 (EXCITEMENT).

References

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Text Analysis Conference*.

Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2010. The sixth PASCAL recognizing textual entailment challenge. In *Proceedings of the Text Analysis Conference*.

Peter Clark and Phil Harrison. 2010. BLUE-Lite: a knowledge-based lexical entailment system for RTE6. In *Proceedings of the Text Analysis Conference*.

Hang Cui, Renxu Sun, Keya Li, Min yen Kan, and Tat seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of SIGIR*.

Abdessamad Echihabi and Daniel Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of ACL*.

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

Nizar Habash and Bonnie Dorr. 2003. A categorial variation database for english. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Aria Haghighi, Andrew Ng, and Christopher Manning. 2005. Robust textual inference via graph matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Houping Jia, Xiaojiang Huang, Tengfei Ma, Xiaojun Wan, and Jianguo Xiao. 2010. PKUTM participation at the Text Analysis Conference 2010 RTE and summarization track. In *Proceedings of the Text Analysis Conference*.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING*.

Andrew MacKinlay and Timothy Baldwin. 2009. A baseline approach to the RTE5 search pilot. In *Proceedings of the Text Analysis Conference*.

Debarghya Majumdar and Pushpak Bhattacharyya. 2010. Lexical based text entailment system for main task of RTE6. In *Proceedings of the Text Analysis Conference*.

Judea Pearl. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Vasin Punyakanok, Dan Roth, and Wen tau Yih. 2004. Mapping dependencies trees: An application to question answering. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*.

Rajat Raina, Andrew Y. Ng, and Christopher D. Manning. 2005. Robust textual inference via learning and abductive reasoning. In *Proceedings of AAAI*.

Stefan Schoenmackers, Oren Etzioni, and Daniel Weld. 2008. Scaling textual inference to the web. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Eyal Shnarch, Jacob Goldberger, and Ido Dagan. 2011a. A probabilistic modeling framework for lexical entailment. In *Proceedings of ACL*.

Eyal Shnarch, Jacob Goldberger, and Ido Dagan. 2011b. Towards a probabilistic model for lexical entailment. In *Proceedings of the TextInfer Workshop on Textual Entailment*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of WWW*.

Mengqiu Wang and Christopher Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of Coling*.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.