PLIS: a Probabilistic Lexical Inference System

Eyal Shnarch[1], Erel Segal-haLevi[1], Jacob Goldberger[2], Ido Dagan[1]
[1]Computer Science Department, Bar-Ilan University, Israel
[2]Faculty of Engineering, Bar-Ilan University, Israel
{shey,erelsgl,dagan}@cs.biu.ac.il
goldbej@eng.biu.ac.il

Abstract

This paper presents PLIS, an open source Probabilistic Lexical Inference System which combines two functionalities: (i) a tool for integrating lexical inference knowledge from diverse resources, and (ii) a framework for scoring textual inferences based on the integrated knowledge. We provide PLIS with two probabilistic implementation of this framework. PLIS is available for download and developers of text processing applications can use it as an off-the-shelf component for injecting lexical knowledge into their applications. PLIS is easily configurable, components can be extended or replaced with user generated ones to enable system customization and further research. PLIS includes an online interactive viewer, which is a powerful tool for investigating lexical inference processes.

1 Introduction and background

Semantic Inference is the process by which machines perform reasoning over natural language texts. A semantic inference system is expected to be able to infer the meaning of one text from the meaning of another, identify parts of texts which convey a target meaning, and manipulate text units in order to deduce new meanings.

Semantic inference is needed for many Natural Language Processing (NLP) applications. For instance, a Question Answering (QA) system may encounter the following question and candidate answer (Example 1):

Q: *which explorer discovered the New World*?
A: *Christopher Columbus revealed America*.

As there are no overlapping words between the two sentences, to identify that A holds an answer for Q, background world knowledge is needed to link *Christopher Columbus* with *explorer* and *America* with *New World*. Linguistic knowledge is also needed to identify that *reveal* and *discover* refer to the same concept.

Knowledge is needed in order to bridge the gap between text fragments, which may be dissimilar on their surface form but share a common meaning. For the purpose of semantic inference, such knowledge can be derived from various resources (e.g. WordNet (Fellbaum, 1998) and others, detailed in Section 2.1) in a form which we denote as *inference links* (often called inference/entailment rules), each is an ordered pair of elements in which the first implies the meaning of the second. For instance, the link *ship→vessel* can be derived from the *hypernym* relation of WordNet.

Other applications can benefit from utilizing inference links to identify similarity between language expressions. In Information Retrieval, the user's information need may be expressed in relevant documents differently than it is expressed in the query. Summarization systems should identify text snippets which convey the same meaning.

Our work addresses a generic, application independent, setting of lexical inference. We therefore adopt the terminology of Textual Entailment (Dagan et al., 2006), a generic paradigm for applied semantic inference which captures inference needs of many NLP applications in a common underlying task: given two textual fragments, termed hypothesis (H) and text (T), the task is to recognize whether T implies the meaning of H, denoted $T \rightarrow H$. For instance, in a QA application, H represents the question, and T a candidate answer. In this setting, T is likely to hold an answer for the question if it entails the question.

It is challenging to properly extract the needed inference knowledge from available resources, and to effectively utilize it within the inference process. The integration of resources, each has its own format, is technically complex and the quality

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 97–102,
Sofia, Bulgaria, August 4-9 2013.

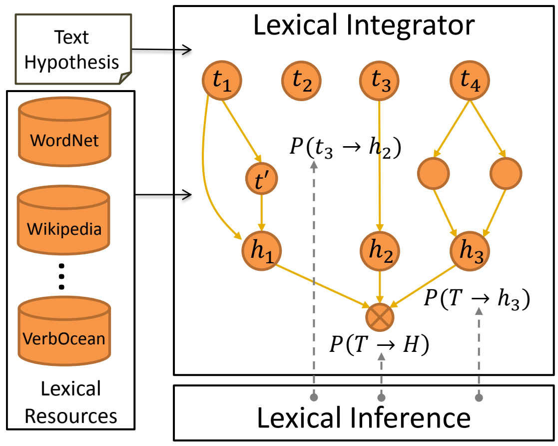

Figure 1: PLIS schema - a text-hypothesis pair is processed by the *Lexical Integrator* which uses a set of lexical resources to extract inference chains which connect the two. The *Lexical Inference* component provides probability estimations for the validity of each level of the process.

of the resulting inference links is often unknown in advance and varies considerably. For coping with this challenge we developed PLIS, a Probabilistic Lexical Inference System[1]. PLIS, illustrated in Fig 1, has two main modules: the *Lexical Integrator* (Section 2) accepts a set of lexical resources and a text-hypothesis pair, and finds all the lexical inference relations between any pair of text term t_i and hypothesis term h_j, based on the available lexical relations found in the resources (and their combination). The *Lexical Inference* module (Section 3) provides validity scores for these relations. These term-level scores are used to estimate the sentence-level likelihood that the meaning of the hypothesis can be inferred from the text, thus making PLIS a complete lexical inference system.

Lexical inference systems do not look into the structure of texts but rather consider them as bag of terms (words or multi-word expressions). These systems are easy to implement, fast to run, practical across different genres and languages, while maintaining a competitive level of performance.

PLIS can be used as a stand-alone efficient inference system or as the lexical component of any NLP application. PLIS is a flexible system, allowing users to choose the set of knowledge resources as well as the model by which inference is done. PLIS can be easily extended with new knowledge resources and new inference models. It comes with a set of ready-to-use plug-ins for many common lexical resources (Section 2.1) as well as two implementation of the scoring framework. These implementations, described in (Shnarch et al., 2011; Shnarch et al., 2012), provide probability estimations for inference. PLIS has an interactive online viewer (Section 4) which provides a visualization of the entire inference process, and is very helpful for analysing lexical inference models and lexical resources usability.

2 Lexical integrator

The input for the lexical integrator is a set of lexical resources and a pair of text T and hypothesis H. The lexical integrator extracts lexical inference links from the various lexical resources to connect each text term $t_i \in T$ with each hypothesis term $h_j \in H$[2]. A lexical inference link indicates a semantic relation between two terms. It could be a directional relation (*Columbus*$\rightarrow$*navigator*) or a bidirectional one (*car* $\longleftrightarrow$ *automobile*).

Since knowledge resources vary in their representation methods, the lexical integrator wraps each lexical resource in a common plug-in interface which encapsulates resource's inner representation method and exposes its knowledge as a list of inference links. The implemented plug-ins that come with PLIS are described in Section 2.1. Adding a new lexical resource and integrating it with the others only demands the implementation of the plug-in interface.

As the knowledge needed to connect a pair of terms, t_i and h_j, may be scattered across few resources, the lexical integrator combines inference links into *lexical inference chains* to deduce new pieces of knowledge, such as *Columbus* $\xrightarrow{resource1}$ *navigator* $\xrightarrow{resource2}$ *explorer*. Therefore, the only assumption the lexical integrator makes, regarding its input lexical resources, is that the inferential lexical relations they provide are transitive.

The lexical integrator generates lexical inference chains by expanding the text and hypothesis terms with inference links. These links lead to new terms (e.g. *navigator* in the above chain example and t' in Fig 1) which can be further expanded, as all inference links are transitive. A *transitivity*

[1]The complete software package is available at http://www.cs.biu.ac.il/nlp/downloads/PLIS.html and an online interactive viewer is available for examination at http://irsrv2.cs.biu.ac.il/nlp-net/PLIS.html.

[2]Where i and j run from 1 to the length of the text and hypothesis respectively.

limit is set by the user to determine the maximal length for inference chains.

The lexical integrator uses a graph-based representation for the inference chains, as illustrates in Fig 1. A node holds the lemma, part-of-speech and sense of a single term. The sense is the ordinal number of WordNet sense. Whenever we do not know the sense of a term we implement the most frequent sense heuristic.[3] An edge represents an inference link and is labeled with the semantic relation of this link (e.g. *cytokine→protein* is labeled with the WordNet relation *hypernym*).

2.1 Available plug-ins for lexical resources

We have implemented plug-ins for the following resources: the English lexicon WordNet (Fellbaum, 1998)(based on either JWI, JWNL or extJWNL java APIs[4]), CatVar (Habash and Dorr, 2003), a categorial variations database, Wikipedia-based resource (Shnarch et al., 2009), which applies several extraction methods to derive inference links from the text and structure of Wikipedia, VerbOcean (Chklovski and Pantel, 2004), a knowledge base of fine-grained semantic relations between verbs, Lin's distributional similarity thesaurus (Lin, 1998), and DIRECT (Kotlerman et al., 2010), a directional distributional similarity thesaurus geared for lexical inference.

To summarize, the lexical integrator finds all possible inference chains (of a predefined length), resulting from any combination of inference links extracted from lexical resources, which link any t, h pair of a given text-hypothesis. Developers can use this tool to save the hassle of interfacing with the different lexical knowledge resources, and spare the labor of combining their knowledge via inference chains.

The lexical inference model, described next, provides a mean to decide whether a given hypothesis is inferred from a given text, based on weighing the lexical inference chains extracted by the lexical integrator.

3 Lexical inference

There are many ways to implement an inference model which identifies inference relations between texts. A simple model may consider the number of hypothesis terms for which inference chains, originated from text terms, were found. In PLIS, the inference model is a plug-in, similar to the lexical knowledge resources, and can be easily replaced to change the inference logic.

We provide PLIS with two implemented baseline lexical inference models which are mathematically based. These are two Probabilistic Lexical Models (PLMs), *HN-PLM* and *M-PLM* which are described in (Shnarch et al., 2011; Shnarch et al., 2012) respectively.

A PLM provides probability estimations for the three parts of the inference process (as shown in Fig 1): the validity probability of each inference chain (i.e. the probability for a valid inference relation between its endpoint terms) $P(t_i \rightarrow h_j)$, the probability of each hypothesis term to be inferred by the entire text $P(T \rightarrow h_j)$ (term-level probability), and the probability of the entire hypothesis to be inferred by the text $P(T \rightarrow H)$ (sentence-level probability).

HN-PLM describes a generative process by which the hypothesis is generated from the text. Its parameters are the reliability level of each of the resources it utilizes (that is, the prior probability that applying an arbitrary inference link derived from each resource corresponds to a valid inference). For learning these parameters *HN-PLM* applies a schema of the EM algorithm (Dempster et al., 1977). Its performance on the recognizing textual entailment task, RTE (Bentivogli et al., 2009; Bentivogli et al., 2010), are in line with the state of the art inference systems, including complex systems which perform syntactic analysis. This model is improved by *M-PLM*, which deduces sentence-level probability from term-level probabilities by a Markovian process. PLIS with this model was used for a passage retrieval for a question answering task (Wang et al., 2007), and outperformed state of the art inference systems.

Both PLMs model the following prominent aspects of the lexical inference phenomenon: (i) considering the different reliability levels of the input knowledge resources, (ii) reducing inference chain probability as its length increases, and (iii) increasing term-level probability as we have more inference chains which suggest that the hypothesis term is inferred by the text. Both PLMs only need sentence-level annotations from which they derive term-level inference probabilities.

To summarize, the lexical inference module

[3] This disambiguation policy was better than considering all senses of an ambiguous term in preliminary experiments. However, it is a matter of changing a variable in the configuration of PLIS to switch between these two policies.

[4] http://wordnet.princeton.edu/wordnet/related-projects/

Figure 2: PLIS interactive viewer with Example 1 demonstrates knowledge integration of multiple inference chains and resource combination (additional explanations, which are not part of the demo, are provided in orange).

provides the setting for interfacing with the lexical integrator. Additionally, the module provides the framework for probabilistic inference models which estimate term-level probabilities and integrate them into a sentence-level inference decision, while implementing prominent aspects of lexical inference. The user can choose to apply another inference logic, not necessarily probabilistic, by plugging a different lexical inference model into the provided inference infrastructure.

4 The PLIS interactive system

PLIS comes with an online interactive viewer[5] in which the user sets the parameters of PLIS, inserts a text-hypothesis pair and gets a visualization of the entire inference process. This is a powerful tool for investigating knowledge integration and lexical inference models.

Fig 2 presents a screenshot of the processing of Example 1. On the right side, the user configures the system by selecting knowledge resources, adjusting their configuration, setting the transitivity limit, and choosing the lexical inference model to be applied by PLIS.

After inserting a text and a hypothesis to the appropriate text boxes, the user clicks on the *infer* button and PLIS generates all lexical inference chains, of length up to the transitivity limit, that connect text terms with hypothesis terms, as available from the combination of the selected input resources. Each inference chain is presented in a line between the text and hypothesis.

PLIS also displays the probability estimations for all inference levels; the probability of each chain is presented at the end of its line. For each hypothesis term, term-level probability, which weighs all inference chains found for it, is given below the dashed line. The overall sentence-level probability integrates the probabilities of all hypothesis terms and is displayed in the box at the bottom right corner.

Next, we detail the inference process of Example 1, as presented in Fig 2. In this QA example, the probability of the candidate answer (set as the text) to be relevant for the given question (the hypothesis) is estimated. When utilizing only two knowledge resources (WordNet and Wikipedia), PLIS is able to recognize that *explorer* is inferred by *Christopher Columbus* and that *New World* is inferred by *America*. Each one of these pairs has two independent inference chains, numbered 1–4, as evidence for its inference relation.

Both inference chains 1 and 3 include a single inference link, each derived from a different relation of the Wikipedia-based resource. The inference model assigns a higher probability for chain 1 since the *BeComp* relation is much more reliable than the *Link* relation. This comparison illustrates the ability of the inference model to learn how to differ knowledge resources by their reliability.

Comparing the probability assigned by the in-

[5]http://irsrv2.cs.biu.ac.il/nlp-net/PLIS.html

ference model for inference chain 2 with the probabilities assigned for chains 1 and 3, reveals the sophisticated way by which the inference model integrates lexical knowledge. Inference chain 2 is longer than chain 1, therefore its probability is lower. However, the inference model assigns chain 2 a higher probability than chain 3, even though the latter is shorter, since the model is sensitive enough to consider the difference in reliability levels between the two highly reliable *hypernym* relations (from WordNet) of chain 2 and the less reliable *Link* relation (from Wikipedia) of chain 3.

Another aspect of knowledge integration is exemplified in Fig 2 by the three circled probabilities. The inference model takes into consideration the multiple pieces of evidence for the inference of *New World* (inference chains 3 and 4, whose probabilities are circled). This results in a term-level probability estimation for *New World* (the third circled probability) which is higher than the probabilities of each chain separately.

The third term of the hypothesis, *discover*, remains uncovered by the text as no inference chain was found for it. Therefore, the sentence-level inference probability is very low, 37%. In order to identify that the hypothesis is indeed inferred from the text, the inference model should be provided with indications for the inference of *discover*. To that end, the user may increase the transitivity limit in hope that longer inference chains provide the needed information. In addition, the user can examine other knowledge resources in search for the missing inference link. In this example, it is enough to add VerbOcean to the input of PLIS to expose two inference chains which connect *reveal* with *discover* by combining an inference link from WordNet and another one from VerbOcean. With this additional information, the sentence-level probability increases to 76%. This is a typical scenario of utilizing PLIS, either via the interactive system or via the software, for analyzing the usability of the different knowledge resources and their combination.

A feature of the interactive system, which is useful for lexical resources analysis, is that each term in a chain is clickable and links to another screen which presents all the terms that are inferred from it and those from which it is inferred.

Additionally, the interactive system communicates with a server which runs PLIS, in a full-duplex WebSocket connection[6]. This mode of operation is publicly available and provides a method for utilizing PLIS, without having to install it or the lexical resources it uses.

Finally, since PLIS is a lexical system it can easily be adjusted to other languages. One only needs to replace the basic lexical text processing tools and plug in knowledge resources in the target language. If PLIS is provided with bilingual resources,[7] it can operate also as a cross-lingual inference system (Negri et al., 2012). For instance, the text in Fig 3 is given in English, while the hypothesis is written in Spanish (given as a list of lemma:part-of-speech). The left side of the figure depicts a cross-lingual inference process in which the only lexical knowledge resource used is a manually built English-Spanish dictionary. As can be seen, two Spanish terms, *jugador* and *casa* remain uncovered since the dictionary alone cannot connect them to any of the English terms in the text.

As illustrated in the right side of Fig 3, PLIS enables the combination of the bilingual dictionary with monolingual resources to produce cross-lingual inference chains, such as *footballer* $\xrightarrow{hypernym}$ *player* $\xrightarrow{manual}$ *jugador*. Such inference chains have the capability to overcome monolingual language variability (the first link in this chain) as well as to provide cross-lingual translation (the second link).

5 Conclusions

To utilize PLIS one should gather lexical resources, obtain sentence-level annotations and train the inference model. Annotations are available in common data sets for task such as QA, Information Retrieval (queries are hypotheses and snippets are texts) and Student Response Analysis (reference answers are the hypotheses that should be inferred by the student answers).

For developers of NLP applications, PLIS offers a ready-to-use lexical knowledge integrator which can interface with many common lexical knowledge resources and constructs lexical inference chains which combine the knowledge in them. A developer who wants to overcome lexical language variability, or to incorporate background knowledge, can utilize PLIS to inject lex-

[6] We used the socket.io implementation.

[7] A bilingual resource holds inference links which connect terms in different languages (e.g. an English-Spanish dictionary can provide the inference link *explorer→explorador*).

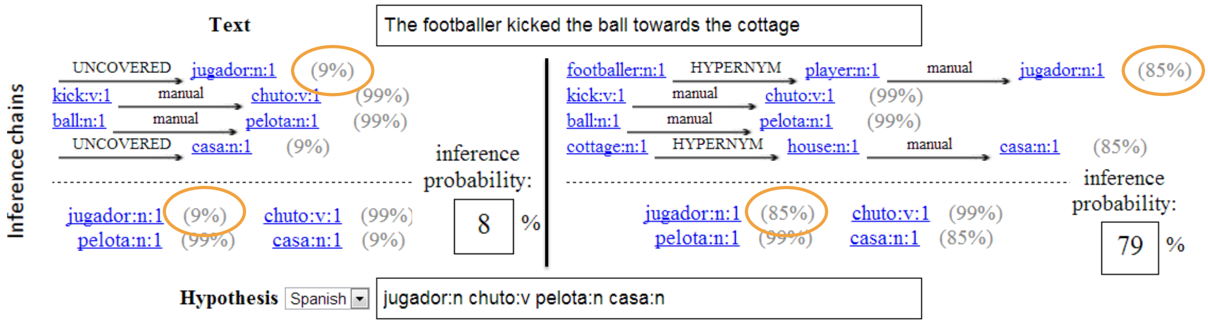

Figure 3: PLIS as a cross-lingual inference system. **Left**: the process with a single manual bilingual resource. **Right**: PLIS composes cross-lingual inference chains to increase hypothesis coverage and increase sentence-level inference probability.

ical knowledge into any text understanding application. PLIS can be used as a lightweight inference system or as the lexical component of larger, more complex inference systems.

Additionally, PLIS provides scores for inference chains and determines the way to combine them in order to recognize sentence-level inference. PLIS comes with two probabilistic lexical inference models which achieved competitive performance levels in the tasks of recognizing textual entailment and passage retrieval for QA.

All aspects of PLIS are configurable. The user can easily switch between the built-in lexical resources, inference models and even languages, or extend the system with additional lexical resources and new inference models.

Acknowledgments

The authors thank Eden Erez for his help with the interactive viewer and Miquel Esplà Gomis for the bilingual dictionaries. This work was partially supported by the European Community's 7^{th} Framework Programme (FP7/2007-2013) under grant agreement no. 287923 (EXCITEMENT) and the Israel Science Foundation grant 880/12.

References

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proc. of TAC*.

Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2010. The sixth PASCAL recognizing textual entailment challenge. In *Proc. of TAC*.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proc. of EMNLP*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Lecture Notes in Computer Science*, volume 3944, pages 177–190.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society, series [B]*, 39(1):1–38.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts.

Nizar Habash and Bonnie Dorr. 2003. A categorial variation database for English. In *Proc. of NAACL*.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proc. of COLOING-ACL*.

Matteo Negri, Alessandro Marchetti, Yashar Mehdad, Luisa Bentivogli, and Danilo Giampiccolo. 2012. Semeval-2012 task 8: Cross-lingual textual entailment for content synchronization. In *Proc. of SemEval*.

Eyal Shnarch, Libby Barak, and Ido Dagan. 2009. Extracting lexical reference rules from Wikipedia. In *Proc. of ACL*.

Eyal Shnarch, Jacob Goldberger, and Ido Dagan. 2011. Towards a probabilistic model for lexical entailment. In *Proc. of the TextInfer Workshop*.

Eyal Shnarch, Ido Dagan, and Jacob Goldberger. 2012. A probabilistic lexical model for ranking textual inferences. In *Proc. of *SEM*.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? A quasi-synchronous grammar for QA. In *Proc. of EMNLP*.