Enabling Transitivity for Lexical Inference on Chinese Verbs Using Probabilistic Soft Logic

Wei-Chung Wang, Lun-Wei Ku
Academia Sinica
128 Academia Road, Section2
Nankang, Taipei 11529, Taiwan
{anthonywang, lwku}@iis.sinica.edu.tw

Abstract

To learn more knowledge, enabling transitivity is a vital step for lexical inference. However, most of the lexical inference models with good performance are for nouns or noun phrases, which cannot be directly applied to the inference on events or states. In this paper, we construct the largest Chinese verb lexical inference dataset containing 18,029 verb pairs, where for each pair one of four inference relations are annotated. We further build a probabilistic soft logic (PSL) model to infer verb lexicons using the logic language. With PSL, we easily enable transitivity in two layers, the observed layer and the feature layer, which are included in the knowledge base. We further discuss the effect of transitives within and between these layers. Results show the performance of the proposed PSL model can be improved at least 3.5% (relative) when the transitivity is enabled. Furthermore, experiments show that enabling transitivity in the observed layer benefits the most.

1 Introduction

Lexical inference is an important component of natural language understanding for NLP tasks such as textual entailment (Garrette et al., 2011), metaphor detection (Mohler et al., 2013), and text generation (Biran and McKeown, 2013) to acquire implications not explicitly written in context. Given two words, the goal of lexical inferences is to detect whether there is an inference relation between the lexicon pair. For example, the word 'buy' entails the word 'have'. With the help of lexical inference system, we can know "Mom has apples" from the ground truth "Mom buys apples"to answer the question "Who has apples?" without explicitly mentioning it.

An intuitive solution to this problem is to first represent the sense of words in the lexicon to calculate the confidence of inferences from one sense to another, or to build a classifier to distinguish inference relations from other relations. Most related research is of one of these two types (Szpektor and Dagan, 2008a; Kiela et al., 2015). However, for this problem it is difficult for these models to take into account transitivity. In the framework of a lexical inference system, transitivity can be included in three layers: the observed layer, the feature layer, and the prediction layer. Figure 1 illustrates these layers and the corresponding transitives. The observed layer includes inference relations we already know, e.g., true inferences from the gold labels or ontologies; the feature layer includes the observed features for all lexicon pairs to be predicted,i.e.,features for the testing data, and the predicted layer saves the predicted inference pairs, i.e., the relations of pairs in the testing data, predicted by the model. As inference usually involves available knowledge, the knowledge base (KB) is shown in Figure 1 as well. KB contains known information for the models. Therefore, in this system, it includes the observed layer and the feature layer which contain gold relations and the features for the testing data respectively.

There has been several new rising research directions involving lexical inference. The most representative ones are the automatic problem solvers and the open-domain question answering systems, where inferring between events or states like *Some animals grow thick fur* effecting *Some animals stay warm* is critical (Clark et al., 2016). However, many recent works of lexical inference are only designed for or being tested on nouns or noun phrases (Jiang and Conrath, 1997; Kiela et al.,

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 110–119,
Taipei, Taiwan, November 27 – December 1, 2017

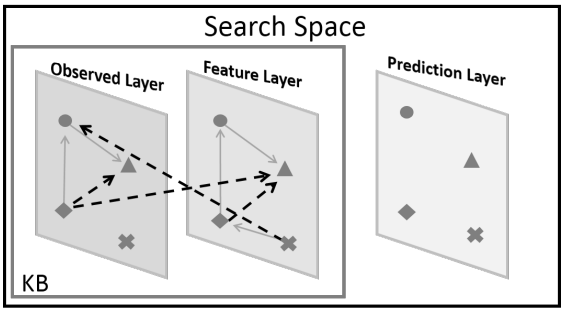

Figure 1: Three-layer lexical inference system. Points of the same shape in each layer are the same verbs; the solid arrow indicates the known inference relation; the dotted arrow indicates the hidden inference relation which can be inferred by the known inference relations.

2015; Shwartz et al., 2016), which makes them limited or not capable for these newly proposed research problems.

In this paper, we adopt the probabilistic soft logic (PSL) model to find lexical inference on Chinese verbs toward the math word problem solver. The contributions of this paper are listed as follows: (1) We build the largest Chinese verb lexical inference dataset with four types of inference relations as a potential testbed in the future. (2) We show that in the proposed PSL model the transitivity is easy to enabled and can benefit the lexical inference on Chinese verbs. (3) We implement and discuss the transitivity inter- and intra- layers and conclude the transitivity within the observed layer brings the most performance gain.

2 Related Work

One mainstream lexical inference extracts either explicit or implicit features from the manually constructed lexical knowledge. Szpektor (2009) constructs a WordNet inference chain through substitution relations (synonyms and hypernyms) defined in WordNet. Aharon (2010) proposed a FrameNet Entailment-rule Derivation (FRED) algorithm to inference on the framework of FrameNet. FrameNet models the semantic argument structure of predicates in terms of prototypical situation, which is called *frames*. Predicates belong to the same *frames* are highly related to a specific situation defined for the frame. Therefore, it is intuitive to acquire lexical inference pairs from predicates in the same frame. However, no matter WordNet or FrameNet was used, the coverage problem was always an issue when leveraging handcraft resources. Moreover, the relations of verbs in WordNet are rather flat compared to nouns, which brings problems when directly adopting approaches utilizing WordNet to detect the inference between verbs.

An unsupervised concept, distributional similarity, for measuring relations between words was proposed to overcome the coverage problem. Distributional similarity related algorithms utilized a large, unstructured corpus to learn lexical entailment relations by assuming that semantically similar lexicons appear with similar context (Harris, 1954). Various implementations were proposed to assess contextual similarity between two lexicons, including (Berant et al., 2010; Lin and Pantel, 2001; Weeds et al., 2004). Lin Similarity, or known as DIRT, is one commonly adopted method to measure the lexical context similarity (Lin and Pantel, 2001). Instead of applying the Distributional Hypothesis to verbs, Lin applied this hypothesis to the paths in dependency trees. They hypothesize that the meaning of two phrases is similar, if their paths tend to link the same sets of words in a dependency tree. Later, Weeds and Weir (2004) proposed a general framework for directional similarity measurement. The measurement examined the coverage of word w_l's features against those of w_r's, and more coverage indicated more similarity.

Lin Similarity generates errors as its symmetric structure cannot tell the difference between $w_l \rightarrow w_r$ and $w_r \rightarrow w_l$. That is, it makes errors on non-symmetric examples, like $buy \rightarrow take$. Moreover, Weeds' method generates high score when an infrequent lexicon has features similar to those of another lexicon, which harms the performance as it happens a lot for non-entailed lexicons. Therefore, Szpektor and Dagan (2008a) proposed a hybrid method **Balanced-Inclusion, BInc**, and it was proved to outperform methods proposed prior to it. In this paper, we adopt BInc measurement and complement with lexical resource method to construct a hybrid model, which was proved to outperform both methods separately on our dataset.

Recent research is exploiting the effect of transitivity during model training. The intuition is that some implicit entailment relation is difficult to be identified when there is no direct features supporting it. Sometimes previous work could find the

entailment pairs $w_1 \rightarrow w_2$ and $w_2 \rightarrow w_3$, but failed to answer distant entailment relation like $w_1 \rightarrow w_3$. Skeptor and Dagan (2009) first applied transitive chaining in the knowledge provided by the lexical ontology Wordnet (Miller, 1995) in the feature layer. Berant et al. (2011) built a lexical entailment knowledge graph given the predicted results from the base classifier. They used integer linear programming (ILP) to find the latent entailment in the prediction cascade, which transits in the prediction layer. Kloetzer et al. (2015), whose system outperformed Berant et al.'s on their own corpus, further use cascade entailment inference in the feature layer. They applied short transitivity optimization by a two-layered SVM classifier (Kloetzer et al., 2015). A set of candidate transitivity paths were created by concatenating two identified inference pairs from the first SVM classifier, e.g., $w_1 \rightarrow w_2$ and $w_2 \rightarrow w_3$ result in a candidate path $w_1 \rightarrow w_2 \rightarrow w_3$. Then the two-layered SVM classifier re-predicted whether there was an inference relation for the lexical pair $w_1 \rightarrow w_3$. However, none of these models takes into account transitivity in the observed layer or transitivity between two layers.

We select probabilistic soft logic (PSL) to model the lexical inference problem. PSL is a recently proposed alternative framework for probabilistic logic (Bach et al., 2015). It was first applied to the category prediction and similarity propagation on Wikipedia documents to align ontologies on a standard corpus of bibliographic ontology (Brocheler et al., 2012). It has been adopted in social network analysis, including social group modeling (Huang et al., 2012) and social trust analysis (Huang et al., 2013). For natural language processing, recently, Dhanya Sridhar (2014) applied the PSL model to stance classification of on-line debates. Islam Beltagy (2014) approached the textual problem by transforming sentences into their logic representations and applying a PSL model to analyze word-to-word semantic coverage between the hypothesis and the premise. All these show that PSL is good at capturing relations. However, PSL has not been utilized yet in the lexical inference problem, and its power to provide lexical transitivity has not been tested, either. Thus in this paper, we explore its ability on detecting verb lexical inference and on enabling the transitivity.

3 Approach

We start from describing the features for each lexicon pair. To use PSL, we define atoms and design rules to enable the inter- and intra-layer transitives. Finally, PSL will automatically learn the rule weights by MLE to yield the best results.

3.1 Lexicon Pair Features

3.1.1 Lexical ontology features

E-HowNet is a large Chinese lexical resource extended from HowNet (Dong and Dong, 2006). Manually constructed by several linguistic experts, it contains 93,953 Chinese words and 9,197 semantic types (concepts; some are sememes). It was designed as an ontology of semantic types, each is listed in both Chinese and in English. For example, one semantic type is $(Give|給)$. Each semantic type has some instances which inherit the concept of it. Lexical relations are also defined. In addition to hypernym-hyponym pairs, E-Hownet contains conflation pairs, including preconditions like (*Divorce*|離婚) is to (*GetMarried*|結婚), consequences like (*Labor*|臨產) is to (*Pregnant*|懷孕), and same-events like (*Sell*|賣) is to (*Buy*|買). The hypernym-hyponym relation and the conflation relation are two features that we use to represent a lexicon pair.

3.1.2 Cohesion path score

Given two semantically related words, a key aspect of detecting lexical inference is the generality of the hypothesis compared to the premise. Though we have a lexical ontology to tell us explicitly the hypernym-hyponym relations, a score to estimate the degree of this compared generality is still necessary for model learning. Therefore, We define the cohesion score of a semantic type with E-Hownet to model the generality. For each semantic type $s_i \in S$ which has a set of instantiate words V_{si}, the cohesion score of s_i is calculated as

$$Coh\,(s_i) = \frac{1}{N} \sum_{v_1 \neq v_2} sim\,(v_1, v_2)\,; \quad v_1, v_2 \in V_{s_i} \tag{1}$$

where $sim(v_1, v_2)$ is the word-embedding cosine similarity of words v_1 and v_2.

We construct a graph by considering hypernym, hyponym, and conflation relations in E-HowNet where nodes are semantic types and instantiate words, and where edges are relations. Given a word pair (v_l, v_r), a set of paths P from v_l to

v_r can be found by traversing this graph, each of which is denoted as p with edges in the edge set E. Each of these edges in E is represented by the triple $e(n_1, n_2, type_e)$, where node n_2 is of type $type_e$ to node n_1. Nodes here can be a word or a semantic type. The $PathScore(p)$ is defined as:

$$PathScore(p) = \prod_{e \in E_p} \begin{cases} coh(s_e), type_e = \text{Hyponym} \\ 1, \text{otherwise} \end{cases} \quad (2)$$

The idea of $PathScore(p)$ is to calculate the generality lost, which is caused by hyponym relations, of each step of inference. The hypernym or conflation relation does not lose generality, so the $PathScore(p)$ is always 1.

Empirically, those path p whose length exceed 10 are dropped as the inference chain is too long. Finally, the cohesion path score of word pair (v_1, v_2) is defined as:

$$CohPathScore(v_1, v_2) = \frac{\ln(\max_{p \in P} PathScore(p)) - \ln(m)}{\ln(M) - \ln(m)} \quad (3)$$

while M and m are the Maximum and Minimum PathScore respectively. The cohesion path score also serves as a feature to build the PSL model.

3.1.3 Distributional similarity

Distributional semantics has been used to exploit the semantic similarities of the linguistic items through large language data.

We applied the CKIP parser [1], a well-known Chinese text parser, to raw sentences. Context of words are extracted as features fs of words, according to parsed sentence trees.

Some pre-prosessing steps are performed. Words appearing only once in the corpus are dropped to reduce Chinese segmentation error. For each Word v, we retrieve all the words that share at least one feature with w and call them candidate words. Drop the candidate word if it shares less than 1 percent features, counted by frequency, with word w. We then calculate the distributional similarity score between w and its candidate words.

Balanced-inclusion (BInc, (Szpektor and Dagan, 2008a)) is a well-known scoring function for determining lexical entailment. It contains two parts, one is semantic similarity measurement, and one is semantic coverage direction measurement. Given two words w_l, w_r and their feature sets F_l, F_r, the semantic similarity between w_l and w_r is calculated by Lin similarity (Lin and Pantel, 2001):

$$Lin(v_l, v_r) = \frac{\sum_{f \in F_l \cap F_r} [w_{vl}(f) + w_{vr}(f)]}{\sum_{f \in F_l} w_{vl}(f) + \sum_{f \in F_r} w_{vr}(f)} \quad (4)$$

The coverage direction measurement, which provides clues of direction of entailment relation, is calculated by Weed's (Weeds et al., 2004) coverage measurement:

$$weed(v_l, v_r) = \frac{\sum_{f \in F_l \cap F_r} w_{vl}(f)}{\sum_{f \in F_l} w_{vl}(f)} \quad (5)$$

The weight of each feature $w(f)$ is the Pointwise Mutual Information (PMI) between the word v and the feature f :

$$w_v(f) = \log[\frac{pr(f|v)}{pr(f)}] \quad (6)$$

where $pr(f)$ is probability of feature f. BInc is defined as geometric mean of the above two:

$$BInc(v_l, v_r) = \sqrt{Lin(v_l, v_r) \cdot Weed(v_l, v_r)} \quad (7)$$

To compare BInc's performance to the proposed PSL model and utilize it as a feature, we implemented it on the Chinese experimental dataset to calculate the BInc score of each lexicon pair.

3.1.4 Word Embeddings

Previous work has shown that word embeddings work well on entailment relation recognition of noun-noun pairs and (adj+noun)-noun pairs (Baroni et al., 2012; Roller et al., 2014). We choose glove (Pennington et al., 2014) to train embeddings of each word, and concatenate the embedding of two words to create the embedding for each word pair. This embedding then serves as the feature in the rbf-kernel SVM classifier to predict the entailment relation of the corresponding word pair.

3.2 Probabilistic Soft Logic (PSL)

We use the PSL model to find the latent inference relations by enabling the transitivity of lex-

[1] CKIP parser : http://parser.iis.sinica.edu.tw/

ical relations. The lexical relations include features described in Section 3.1, and the known inference relations in the observed layer. In PSL, each relation of the lexicon pair v_l, v_r is written as a (ground) atom $a(v_l, v_r)$ in the logic language. The description of the transitivity of atoms $a_i(v_1, v_2)$, $a_j(v_2, v_3)$ and its latent inference relation, $Etl(v_1, v_3)$ is written as a rule in the logic language:

$$a_i(v_1, v_2) \wedge a_j(v_2, v_3) \rightarrow Etl(v_1, v_3) \quad (8)$$

Each rule is assigned a weight to denote the reliability of the hypothesis that given $a_i(v_1, v_2)$, $a_j(v_2, v_3)$ are true, $Etl(v_1, v_3)$ is also true. The PSL model learns the rule weights by the training set. We encode the transitivity inter-($i = j$) and intra-($i \neq j$) different types of relations and their resulting latent inference relation to construct the experimental rule set.

Given a set of (ground) atoms $a = \{a_1, ..., a_n\}$, we denote an *interpretation* the mapping $I : a \rightarrow [0, 1]^n$ from ground atoms to soft truth value. The *distance to satisfaction* of each ground rule is defined as:

$$d(r, I) = \max\{0, I(r_{antecedent}) - I(r_{consequent})\} \quad (9)$$

The PSL model learns the weights λ_r of these rules and optimizes the most probable *interpretation* of entailment relations, through the probability density function f over I:

$$f(I) = \frac{1}{Z} \exp[-\sum\nolimits_{r \in R} \lambda_r (d(r, I))^p]; \quad (10)$$

where Z is the normalization term, λ_r is the weight of rule r, R is the set of all ground rules, and $p \in \{1, 2\}$. In this paper, we set p to 2, indicating a squared function.

In the following section, we are going to describe the atoms defined in our lexical inference model in Section 3.2.1. Then rules are defined in Section 3.2.2. Last, weight learning is described in Section 3.2.3

3.2.1 Atoms for PSL

Atoms are types of information provided in Knowledge base in PSL model, Table 1 lists all atoms defined in our lexical inference model. Etl denotes the entailment relation serving as the prediction target. It is the only unknown atom. In PSL model the number of prediction target grows quadratically with the number of the entities (verbs), if no limitation is provided, which is not desired and is time consuming. Thus Cdd indicates canopies (McCallum et al., 2000) over the prediction target. $Hypr$, Con, Coh, and $BInc$ are the hypernym, conflation, cohesion path score, and distributional similarity score BInc features described in Section 3.1. Svm is the prediction of SVM classifier which takes concatenation of word embeddings as feature. Obv represents the knowledge of observed entailment lexical pairs for the training phase. Note that the set of pairs with $Obv = true$ must not overlap with the testing set.

3.2.2 Inference rules for PSL

Having defined the atoms, the five features $Hypr$, Con, $BInc$, Coh, and Svm are used in the design of five basic rules in Eq. 11. We further apply the inference chain by concatenating two atoms to create 25 rules shown as Eq.12 for feature-layer transitivity. For transitivity in the observed layer, we concatenate Obv atoms as shown in Eq.13. Then we concatenate Obv with other features and vice versa to add 10 additional rules shown as in Eq.14,15 for bidirectional transitives between the feature and the observed layers. Finally, the rule $\neg Etl(v_1, v_2)$ states that v_1 does not entail v_2 if the previous rules are not applicable.

$$Rel(v_1, v_2) \rightarrow Etl(v_1, v_2); \quad Rel \in \{Hypr, Con, BInc, Coh, Svm\} \quad (11)$$

$$Rel(v_1, v_2) \wedge Rel(v_2, v_3) \rightarrow Etl(v_1, v_3) \quad (12)$$

$$Obv(v_1, v_2) \wedge Obv(v_2, v_3) \rightarrow Etl(v_1, v_3) \quad (13)$$

$$Obv(v_1, v_2) \wedge Rel(v_2, v_3) \rightarrow Etl(v_1, v_3) \quad (14)$$

$$Rel(v_1, v_2) \wedge Obv(v_2, v_3) \rightarrow Etl(v_1, v_3) \quad (15)$$

3.2.3 Learning inference rule weights

The rule weights(λ_r) are determined using maximum-likelihood estimation.

$$\frac{\partial}{\partial \lambda_r} \log p(I) = -\sum\nolimits_{r \in R_i} (d(r, I)) + E\left[\sum\nolimits_{r \in R_i} (d(r, I))\right] \quad (16)$$

Atom Name	Description
Cdd(v_1,v_2)	Canopies over prediction target. Return 1 if (v_1,v_2) is the prediction target in the task
Etl(v_1,v_2)	Entail statement which is the prediction target.
Hypr(s_1,s_2)	Hypernym relation between two semantic concept: s_1 is hypernym of s_2.
Con(s_1,s_2)	Conflation relation between two semantic types.
Ehow(v_1, v_2)	E-HowNet algorithm.
Dis(v_1,v_2)	BInc between v_1 and v_2.
Svm(v_1,v_2)	Svm prediction featured by word embeddings
Obv(v_1,v_2)	Observed entail statement.

Table 1: List of atoms in lexical inference model

The expected value $E\left[\sum_{r\in R_i}(d(r,I))\right]$ is intractable. Thus it is approximated via $\sum_{r\in R_i} d_r(I^*)$, where I^* is the most probable interpretation given the current weight (Kimmig et al., 2012).

4 Evaluation

4.1 Experiment Dataset

There are some of entailment dataset open to research utility, but the Chinese Verb entailment dataset (CVED) is special in some way. First, most of the open entailment dataset include the entailment between noun-noun pairs, adjective_noun-noun pairs, and quantity_noun-quantity_noun pairs, but none of them consider the entailment between verb-verb pairs like CVED. Second, in my knowledge, our CVED is the largest Chinese entailment dataset.

To get more verb lexical inference pairs for our experiments, we collected verb pairs from math application problems, which usually contain logical relations in the descriptions for each problem. A total of 995 verbs and 18,029 verb pairs were extracted from 20,000 Chinese elementary math problems, where the verbs in each pair are from the same problem. Few types of verb are discarded, including **V_1**, **V_2**, **VH**, **VI**, **VJ**, **VK** and **VL** ,which are adjective[2] and statement associated verbs defined in CKIP[3].

Given a set of verbs extracted from a math problem, every possible directed verb pair was labeled. If there were n verbs, $n \times (n-1)$ directed verb pairs $(v_i \rightarrow v_j)$ were collected, where v_i is the premise and v_j is the hypothesis. For example, if we extracted "sell", "buy", and "pay" from the descriptions of the problem, we added six directed verb pairs to the annotation set: $\{(sell, buy), (sell, pay), (buy, pay), (buy, sell), (pay, sell), (pay, buy)\}$ We provide four types of entailment label in CVED. One is commonly seen hypernym relation. The same-event relations are verb pairs related to same thing but in different point of view Some examples are (sell, buy) and (give, got). These are used by most earlier research or in small-scale experiments (Szpektor and Dagan, 2008b; Kiela et al., 2015). Another two are casual relations, as premises in the *precondition* and *consequence* relations are likely to be true given their hypothesis in our daily life, and because these relations are more useful in real applications, we further consider these relations as entailment relations. These relations are usually selected for web-scale experiments (Aharon et al., 2010; Berant et al., 2011; Kloetzer et al., 2015). Among all experimental verb pairs, 10% were used for testing, 10% were used for developing and the remaining dataset was for training. A five-fold training process was performed to learn the best parameters for the testing model.

4.2 Experiment Setting

To achieve better performance, weights are randomly initialized and retrained 10 times for each fold. The best combination is derived by averaging the five best weight sets obtained in the five-fold cross-validation process. Two baselines are provided for the evaluation of the models with transitivity disabled. Hyper+Conf is the ontology-based baseline. In this setting, verb pairs with hypernym and conflation relations found in E-Hownet are reported as entailment pairs. BInc is the distributional similarity baseline, where we set a best threshold for the development set and apply it

[2] Adjective words are seen as kind of verbs in CKIP

[3] http://rocling.iis.sinica.edu.tw/CKIP/tr/9305_2013%20revision.pdf

	Precision	Recall	F1
Hyper+Conf	**0.547**	0.189	0.281
BInc	0.150	0.098	0.119
PSL	0.270	**0.474**	**0.344**

Table 2: Model performance: transitivity disabled.

to the testing set to identify the entailment relation. The 20,000 elementary math problems together with 61,777 sentences from Sinica Treebank[4] are utilized to calculate the BInc score of each verb pair. A set of 300 dimensional word embedding representation is trained by a hybrid of Sinica Treebank, elementary math problems and Chinese Wikipedia.

To discuss the effect of transitivity within (intra-) and between (inter-) different layers, we build three additional models for PSL. PSL_TrFeat allows transitivity within the feature layer, PSL_TrObv allows transitivity within the observed layer on top of PSL_TrFeat, and PSL_TrFeatObv allows transitivity betwen the observed layer and the feature layer on top of PSL_TrObv. Here we set the degree of transitivity to 2, and leave the determination of the best transitivity degree as future work. For comparison, we implement a SVM baseline ,the state-of-the-art entailment classifier (Kloetzer(base)), and its transitivity framework (Kloetzer(TrFeatPred)) (Kloetzer et al., 2015). We use rbf-kernel SVM and the other hyper-parameters are selected from the 5-fold training.

4.3 Results and Discussion

Table 2 shows the performance of the proposed PSL model when transitivity is disabled (PSL). Unsurprisingly, Hyper+Conf achieves the highest precision as the relations found in E-Hownet are built manually. False alarms come from pairs that contain various unknown Chinese compound words that E-Hownet does not include, e.g., 分給(distribute to) is composed of 分(issue) and 給(give). We attempt to find its head to determine its sense, which sometimes causes errors. Compared to BInc, though in general distributional approaches may outperform ontology-based approaches at least in recall, Hyper+Conf still performs much better. We think the reason is that E-Hownet already contains a large number of words and adopting the heuristic of finding the head for compound words which could mitigates the coverage problem.

Table 3 shows the performance of various PSL models when transitivity is enabled. We conduct a SVM baseline, SVM(w2v), by concatenating the word embeddings of two verbs as the features of the verb pair and it performs comparably well, indicating word embeddings are strong features. Therefore, we discuss the effect of the strong and the weak base settings here. The strong base setting involves the prediction of SVM by word embeddings (relation *SVM*), while the weak base setting involves the rest relations *Hypr*, *Con*, *BInc* and *Coh*. The SVM model from Kloetzer serves as the second baseline. It involves more than 100 features but does not include word embeddings, and hence we compare it with the PSL models of the weak base setting. For the weak base setting, the performance of PSL cannot beat that of Kloetzer's SVM in the very beginning, as SVM is generally considered a more powerful classifier and the Kloetzer's SVM model involves comparably more features. Surprisingly, this state-of-the-art model from Kloetzer does not improve its F1 score after enabling the transitivity in the feature layer by their transitivity framework. (Kloetzer(TrFeatPred) vs. Kloetzer(base): they report a 2% improvement in average precision in their paper.) For the proposed PSL models, enabling transitivity in the feature layer (PSL(TrFeat) vs. PSL(base)) does improve the F1 score from the gain of recall. The reason for this could be that the transitivities of Kloetzer's features depend on the transitivities of the prediction results. If the predictions don't indicate a path to transit, their features will not be combined together for the next prediction. Therefore, their transitivity framework may involve the noise from the first prediction. On the contrary, in our PSL models, all possible feature-layered transitivities between pairs are explored. Hence, our feature-layered transitivity models have the capabilities to improve the recall.

A significant improvement comes from enabling transitivity in the observed layer, that is, if we know $w_1 \rightarrow w_2$ and $w_2 \rightarrow w_3$, we add $w_1 \rightarrow w_3$ to the gold labels. As the relations in the observed layer constitute prior knowledge (known from the training data and saved in the PSL knowledge base), inferring from one relation to the other involves less uncertainty. Therefore, compared to

[4]sinica treeback: http://rocling.iis.sinica.edu.tw/CKIP/engversion/treebank.htm

	Precision	Recall	F1
SVM(w2v)	0.850	0.500	0.630
PSL(WeakBase)	0.314	0.570	0.405
PSL(WeakBase_TrFeat)	0.348	0.645	0.452
PSL(WeakBase_TrObv)	0.675	0.577	0.622
PSL(WeakBase_TrFeatObv)	0.544	0.613	0.577
Kloetzer(base)	0.390	0.590	0.469
Kloetzer(TrFeatPred)	0.385	0.604	0.470
PSL(StrongBase)	0.670	0.649	0.660
PSL(StrongBase_TrFeat)	0.667	0.649	0.658
PSL(StrongBase_TrObv)	0.624	0.757	**0.684**
PSL(StrongBase_TrFeatObv)	0.612	**0.764**	0.680

Table 3: Model performance: transitivity enabled. PSL(StrongBase_TrObv) is significantly better than all the other models with p-value < 0.001.

PSL(WeakBase_TrFeat), PSL(WeakBase_TrObv) shows a great improvement in both precision and F1. For recall, the feature-layer transitivity (PSL(WeakBase_TrFeat)) enables the model to reach more words for a better recall, while the enrichment of the prior knowledge in PSL(WeakBase_TrObv) helps to eliminate uncertainty but decreases recall. If we go further to enable transitivity between the observed layer and the feature layer using model PSL(WeakBase_TrFeatObV), it begins to suffer from the lower precision caused by longer transitivity. Overall, PSL(WeakBase_TrObV) achieves best among all PSL(WeakBase) models, with improvements of 21.7% over the transitivity-disabled PSL model.

Compared to the models of the weak base setting, the PSL model of the strong base setting without transitivity enabled has achieved good performance in the very beginning (F1=0.66). Its performance is better than 3 baselines, SVM(w2v), Kloetzer(base) and Kloetzer(TrFeatPred). It also performs better than the best PSL model of the weak base setting, PSL(WeakBase_TrObv). The great thing is, enabling transitivity achieves even better performance in PSL(StrongBase_TrObv) and PSL(StrongBase_TrFeatObv). For all models of the strong base settings, only enabling the transitivity in the feature layer does not benefit the performance as this decreases the precision.

From all the experiment results, we can conclude the followings. First, enabling transitivities help to find more inference pairs no matter the initial model is strong or weak. Second, for a general model, transitivities inter- or intra- layers both help it become stronger; however, for a strong model, only the transitivities intra- or inter the observed layer, i.e., involving the gold labels, contribute to the performance gain. In other words, only solid knowledge can make a strong model even stronger through transitivities.

5 Conclusion

We have proposed a PSL model to explore the power of transitivity. In this process, the easy and straightforward nature of PSL in considering transitives for lexical inference is demonstrated. Results show that the best PSL model achieves the F1 score 0.684. Moreover, the proposed base PSL model has already achieved well and models with transitivity enabled achieve even better, which confirms the power of transitivity for solving the lexical inference problem on verbs. We will release the current experimental dataset. Future goals include enlarging our dataset by including web word pairs and applied the predicted results in textual entailment tasks. The constructed CVED dataset can be found in the NLPSA lab webpage[5].

Acknowledgments

Research of this paper was partially supported by Ministry of Science and Technology, Taiwan, under the contract 105-2221-E-001-007-MY3.

[5]http://academiasinicanlplab.github.io/

References

Roni Ben Aharon, Idan Szpektor, and Ido Dagan. 2010. Generating entailment rules from framenet. In *Proceedings of the ACL 2010 Conference Short Papers*. Association for Computational Linguistics, pages 241–246.

Stephen H Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2015. Hinge-loss markov random fields and probabilistic soft logic. *arXiv preprint arXiv:1505.04406* .

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 23–32.

Islam Beltagy, Katrin Erk, and Raymond J Mooney. 2014. Probabilistic soft logic for semantic textual similarity. In *ACL (1)*. pages 1210–1219.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2010. Global learning of focused entailment graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1220–1229.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 610–619.

Rahul Bhagat, Patrick Pantel, Eduard H Hovy, and Marina Rey. 2007. Ledir: An unsupervised algorithm for learning directionality of inference rules. In *EMNLP-CoNLL*. pages 161–170.

Or Biran and Kathleen McKeown. 2013. Classifying taxonomic relations between pairs of wikipedia articles. In *IJCNLP*. pages 788–794.

Matthias Brocheler, Lilyana Mihalkova, and Lise Getoor. 2012. Probabilistic similarity logic. *arXiv preprint arXiv:1203.3469* .

Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Turney, and Daniel Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions .

Zhendong Dong and Qiang Dong. 2006. *HowNet and the Computation of Meaning*. World Scientific.

Dan Garrette, Katrin Erk, and Raymond Mooney. 2011. Integrating logical representations with probabilistic information using markov logic. In *Proceedings of the Ninth International Conference on Computational Semantics*. Association for Computational Linguistics, pages 105–114.

Klir George J and Yuan Bo. 2008. Fuzzy sets and fuzzy logic, theory and applications. - .

Zellig S Harris. 1954. Distributional structure. *Word* 10(2-3):146–162.

Bert Huang, Stephen H Bach, Eric Norris, Jay Pujara, and Lise Getoor. 2012. Social group modeling with probabilistic soft logic. In *NIPS Workshop on Social Network and Social Media Analysis: Methods, Models, and Applications*. volume 7.

Bert Huang, Angelika Kimmig, Lise Getoor, and Jennifer Golbeck. 2013. A flexible framework for probabilistic models of social trust. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, Springer, pages 265–273.

Jay J Jiang and David W Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008* .

Douwe Kiela, Laura Rimell, Ivan Vulic, and Stephen Clark. 2015. Exploiting image generality for lexical entailment detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. ACL.

Angelika Kimmig, Stephen Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2012. A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*. pages 1–4.

Julien Kloetzer, Kentaro Torisawa, Chikara Hashimoto, and Jong-hoon Oh. 2013. Large-scale acquisition of entailment pattern pairs. In *In Information Processing Society of Japan (IPSJ) Kansai-Branch Convention*. Citeseer.

Julien Kloetzer, Kentaro Torisawa, Chikara Hashimoto, and Jong-Hoon Oh. 2015. Large-scale acquisition of entailment pattern pairs by exploiting transitivity .

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering* 7(04):343–360.

Andrew McCallum, Kamal Nigam, and Lyle H Ungar. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 169–178.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

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

Michael Mohler, David Bracewell, David Hinote, and Marc Tomlinson. 2013. Semantic signatures for example-based linguistic metaphor detection. In *Proceedings of the First Workshop on Metaphor in NLP*. pages 27–35.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–43.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *COLING*. pages 1025–1036.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. *arXiv preprint arXiv:1603.06076* .

Dhanya Sridhar, Lise Getoor, and Marilyn Walker. 2014. Collective stance classification of posts in online debate forums. *ACL 2014* page 109.

Idan Szpektor and Ido Dagan. 2008a. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 849–856.

Idan Szpektor and Ido Dagan. 2008b. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 849–856.

Idan Szpektor and Ido Dagan. 2009. Augmenting wordnet-based inference with argument mapping. In *Proceedings of the 2009 Workshop on Applied Textual Inference*. Association for Computational Linguistics, pages 27–35.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, page 1015.