A Family of Fuzzy Orthogonal Projection Models for Monolingual and Cross-lingual Hypernymy Prediction

Chengyu Wang
School of Computer Science and Software Engineering, East China Normal University
chywang2013@gmail.com

Yan Fan
School of Computer Science and Software Engineering, East China Normal University
eileen940531@gmail.com

Xiaofeng He*
School of Computer Science and Software Engineering, East China Normal University
xfhe@sei.ecnu.edu.cn

Aoying Zhou
School of Data Science and Engineering, East China Normal University
ayzhou@dase.ecnu.edu.cn

ABSTRACT

Hypernymy is a semantic relation, expressing the "is-a" relation between a concept and its instances. Such relations are building blocks for large-scale taxonomies, ontologies and knowledge graphs. Recently, much progress has been made for hypernymy prediction in English using textual patterns and/or distributional representations. However, applying such techniques to other languages is challenging due to the high language dependency of these methods and the lack of large training datasets of lower-resourced languages.

In this work, we present a family of fuzzy orthogonal projection models for both monolingual and cross-lingual hypernymy prediction. For the monolingual task, we propose a Multi-Wahba Projection (MWP) model to distinguish hypernymy vs. non-hypernymy relations based on word embeddings. This model establishes distributional fuzzy mappings from embeddings of a term to those of its hypernyms and non-hypernyms, which consider the complicated linguistic regularities of these relations. For cross-lingual hypernymy prediction, a Transfer MWP (TMWP) model is proposed to transfer the semantic knowledge from the source language to target languages based on neural word translation. Additionally, an Iterative Transfer MWP (ITMWP) model is built upon TMWP, which augments the training sets of target languages when target languages are lower-resourced with limited training data. Experiments show i) MWP outperforms previous methods over two hypernymy prediction tasks for English; and ii) TMWP and ITMWP are effective to predict hypernymy over seven non-English languages.

CCS CONCEPTS

• **Computing methodologies** → **Lexical semantics**; *Ontology engineering*; • **Information systems** → *Data mining*;

KEYWORDS

hypernymy prediction, Multi-Wahba Projection, cross-lingual transfer learning

ACM Reference Format:
Chengyu Wang, Yan Fan, Xiaofeng He, and Aoying Zhou. 2019. A Family of Fuzzy Orthogonal Projection Models for Monolingual and Cross-lingual Hypernymy Prediction. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3308558.3313439

1 INTRODUCTION

Hypernymy is a type of basic semantic relations, expressing the "is-a" relation between a concept (hypernym) and its instances (hyponyms). The automatic extraction of hypernymy relations is vital for building large-scale taxonomies, ontologies and knowledge graphs [10, 33, 51]. Such relations also benefit a variety of Web-scale applications including query understanding [48], post-search navigation [14], personalized recommendation [56], etc.

Due to its importance in NLP and real-world applications, remarkable progress has been made for hypernymy prediction. As summarized in Wang et al. [45], pattern based and distributional approaches are two major types of methods for hypernymy harvesting. Pattern based methods employ lexical patterns (e.g., Hearst patterns [13]) to extract hypernymy relations from text corpora [30]. Distributional methods can be divided as unsupervised and supervised. Unsupervised methods refer to hypernymy measures, modeling the degree of the existence of a hypernymy relation between two terms [15, 31]. Supervised algorithms classify a term pair as hypernymy or non-hypernymy based on the distributional representations of the two terms [20, 54]. While there exists some disagreement in the NLP community on which type of methods is more effective [45], Shwartz et al. [34] show that both methods can be combined via an integrated neural network, further improving the performance.

Despite the significant success, there is much room for improvement, due to the following reasons. i) The linguistic regularities of hypernymy relations are complicated to model [10, 44, 53]. For example, semantics of hypernymy relations between concept-concept pairs (subclass-of) and concept-instance pairs (instance-of) are different. Such pairs, however, are often mixed in real-world scenarios.

*Corresponding author.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313439

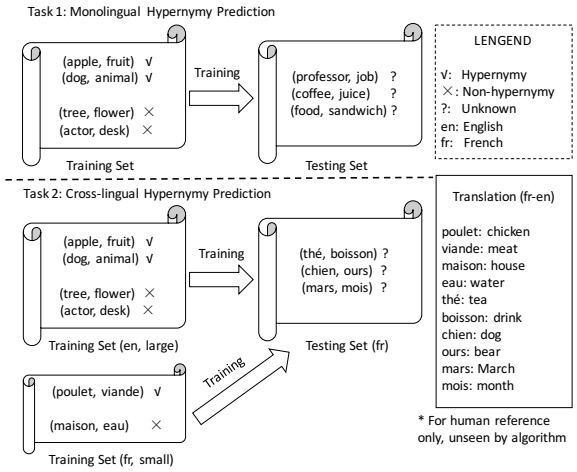


Figure 1: A toy example of monolingual and cross-lingual hypernymy prediction.

ii) Supervised distributional methods are reported to have high performance, but suffer from the "lexical memorization" problem [18]. iii) Most existing methods focus on monolingual hypernymy prediction in English only and are highly language dependent. Some analytical languages (e.g., Chinese, Vietnamese) are flexible in expressions and lack fixed patterns to extract hypernymy relations with high recall [10, 46]. Hence, most existing methods are difficult to apply to other languages. iv) Several distributional methods require additional knowledge (e.g., taxonomies and semantic hierarchies) to learning high-quality hypernymy embeddings [20, 26, 54], which are difficult to obtain for some low-resourced languages.

In this work, a family of fuzzy orthogonal projection models is presented to address the tasks of both monolingual and cross-lingual hypernymy prediction. A toy example of the two tasks is illustrated in Figure 1. For the monolingual task, we explicitly model how the embeddings of a term are mapped to those of its hypernyms or non-hypernyms, respectively, in order to avoid the "lexical memorization" problem [18]. To deal with complicated linguistic regularities, for each latent component of hypernymy and non-hypernymy relations, a fuzzy orthogonal mapping is established in the embedding space[1]. The learning process of the mappings is closely related to a generalized version of the Wahba's problem in mathematics [23], named Multi-Wahba Projection (MWP). In this work, we derive a closed-form solution to MWP and propose an MWP based neural network to distinguish hypernymy vs. non-hypernymy relations.

For cross-lingual hypernymy prediction, the goal is to predict hypernymy relations of a target language with limited training data, given a (relatively) large training set of a source language as input. Neural machine translation techniques are able to transfer knowledge across languages, but usually require a large parallel corpus, which is not feasible for some lower-resourced languages [55]. To derive a general solution for any languages, we present a Transfer MWP (TMWP) model from a source language to target languages based on neural word translation. To handle the small size of training sets of target languages, an Iterative Transfer MWP (ITMWP) model is built upon TMWP to support semi-supervised learning via training data augmentation.

In summary, we make the following major contributions:

- We propose a Multi-Wahba Projection (MWP) model to distinguish monolingual hypernymy vs. non-hypernymy relations. A closed-form solution is derived for generalized Wahba's problem.
- We propose a Transfer Multi-Wahba Projection (TMWP) model for cross-lingual hypernymy prediction. An iterative data augmentation method is introduced, named the Iterative Transfer Multi-Wahba Projection (ITMWP) model.
- We conduct extensive experiments to show that i) MWP outperforms state-of-the-art over two hypernymy prediction tasks for English, and ii) TMWP and ITMWP are effective to predict hypernymy over seven non-English languages.

The rest of this paper is organized as follows. Section 2 summarizes related work. MWP, TMWP and ITMWP models are presented in Section 3 and Section 4. Experiments are shown in Section 5 and Section 6, with the conclusion drawn in Section 7.

2 RELATED WORK

This section presents a brief overview on monolingual and cross-lingual hypernymy prediction.

2.1 Monolingual Hypernymy Prediction

In the NLP community, "hypernymy prediction" is not a standalone task, but has been evaluated under a series of tasks, e.g., unsupervised hypernymy classification [15], supervised hypernymy detection [20, 54], graded lexical entailment [41], etc. Although evaluation steps and metrics may differ, pattern based and distributional methods are two major paradigms to address these tasks [45].

Pattern based approaches use lexical patterns to extract hypernymy relations from texts. Hearst patterns [13] are the most influential patterns in English, which are employed to build a large lexical taxonomy Probase [51]. A recent study [30] reveals that using simple Hearst patterns can result in high performance for hypernymy detection. Similar Hearst pattern-like features are used in a variety of methods [28, 35]. The semantics of more generalized textual patterns can be captured by LSTM-based neural networks [34]. The reason of the success is that they are precise and have high coverage of hypernymy relations in English. For other languages, Hearst-style patterns are not necessarily effective. For example, Chinese has flexible expressions and lacks fixed patterns for hypernymy detection [10, 46]. Because we aim at providing a general solution for all languages, pattern-based approaches are difficult to apply to other languages, without language-specific modifications.

Distributional methods can be divided as supervised and unsupervised. **Unsupervised distributional approaches** are primarily based on hypernymy measures, modeling the degree of the existence of a hypernymy relation within a term pair. Typical hypernymy measures are based on distributional inclusion hypothesis [17], distributional informativeness hypothesis [31] and selective distributional inclusion hypothesis [29]. Readers can refer

[1]For hypernymy, relations related to different domains are usually treated as different components. The "subclass-of" and "instance-of" relations are treated as different components. Components of non-hypernymy relations include co-hyponymy, synonymy, meronymy, etc. Refer to [10, 49, 53] for detailed discussion.

to a comprehensive overview and evaluation on hypernymy measures in [32]. In **supervised methods**, each pair is modeled as an embedding vector as the input of a classifier to predict the relation, such as the Concat model, the Diff model, etc [29, 49]. Recently, several algorithms are proposed to learn hypernymy embeddings, which consider the taxonomic structure of concepts [6, 20, 26, 54]. For example, Yu et al. [54] learn the hypernym and hyponym embeddings for a term in a max-margin neural network based on Probase [51]. Nguyen et al. [26] propose the hierarchical embeddings trained over a text corpus and the WordNet concept hierarchy. Additionally, Drozd et al. [9] study relationships between semantic relations and word embeddings.

Projection based approaches are variants of previous methods, which learn how to map the embeddings of a term to those of its hypernyms directly. A notable model is the piecewise linear projection model [10]. It is also improved by Yamane et al. [53], which learns the number of linear projection models and the parameters of projection models jointly, and by Biemann et al. [3], Wang and He [44], which consider explicit negative samples. Wang et al. [46] learn the representations of both hypernymy and non-hypernymy in a transductive learning framework. Our work improves these approaches by multiple fuzzy orthogonal projections, and also employs a neural network for hypernymy relation classification. Hence, it takes advantages of both classification and projection approaches. Apart from the previous methods, many approaches aim at extracting hypernymy relations from various Web data sources. Gupta et al. [12], Ponzetto and Strube [27], Suchanek et al. [36] learn hypernymy from the Wikipedia category system based on lexical patterns, syntactic constraints and rule-based inference. Liu et al. [19] construct a taxonomy purely from keywords. We do not further elaborate.

2.2 Cross-lingual Hypernymy Prediction

The task of **cross-lingual hypernymy prediction** has not been sufficiently studied. Some researchers aim at building taxonomies based on multi-lingual Wikipedia [21, 47]. YAGO 3 [21] is the extension of YAGO [36], which integrates a multi-lingual taxonomy derived from Wikipedia. Wang et al. [47] propose a cross-lingual knowledge validation technique to improve the accuracy of cross-lingual hypernymy prediction from links among multi-lingual versions of Wikipedia. Wu et al. [50] introduce a bilingual topic model to align taxonomies of different languages.

In the NLP community, a few works [38, 42] determine if a word in one language is a hypernym of a word in another language. For example, these methods predict if the English word "fruit" is the hypernym of the French word "pomme" (apple). The difference between these methods and ours is that our method is capable of learning hypernymy relations where both hypernyms and hyponyms are in the target language. Hence, our method can extract culture-specific hypernymy relations where both hypernyms and hyponyms are not present in the source language.

3 MWP: THE MONOLINGUAL MODEL

In this section, we introduce the MWP model for monolinugal hypernymy prediction and give the closed-form solution of the generalized Wahba's problem.

3.1 Task Definition

Let $\vec{x}_i$ be the normalized embedding of the term x_i, pre-trained using any neural language models. Denote $y_i^{(+)}$ and $y_i^{(-)}$ as a hypernym and a non-hypernym of x_i, respectively. The task of monolingual hypernymy prediction is defined as follows:

Definition 3.1. (Monolingual Hypernymy Prediction) The goal is to train a classifier f over a hypernymy relation set $D^{(+)} = \{(x_i, y_i^{(+)})\}$ and a non-hypernymy relation set $D^{(-)} = \{(x_i, y_i^{(-)})\}$, to predict hypernymy relations $U = \{(x_i, y_i)\}$ of the same language.

3.2 Learning Hypernymy Projections

Following [10, 46, 53], hypernymy projection models assume that there exists a $d \times d$ projection matrix $\mathbf{M}^{(+)}$ such that $\mathbf{M}^{(+)}\vec{x}_i \approx \vec{y}_i^{(+)}$ where d is the dimension of word embeddings and $(x_i, y_i^{(+)}) \in D^{(+)}$. In this work, MWP further forces that the cosine similarity of the projected word embedding $\mathbf{M}\vec{x}_i$ and the true hypernym embedding $\vec{y}_i^{(+)}$ should be maximized, i.e., $\max \sum_{i=1}^{|D^{(+)}|} \cos(\mathbf{M}\vec{x}_i, \vec{y}_i^{(+)})$. Because all the word embeddings are normalized, the optimization function can be re-written as: $\max \sum_{i=1}^{|D^{(+)}|} (\mathbf{M}\vec{x}_i)^T \vec{y}_i^{(+)}$. We can see that $\mathbf{M}$ should be orthogonal to guarantee that $\mathbf{M}\vec{x}_i$ is normalized [52]. For ease of the model training process, we re-formulate the objective function as below with $\mathbf{I}$ being the $d \times d$ identity matrix:

$$\min \sum_{i=1}^{|D^{(+)}|} \|\mathbf{M}\vec{x}_i - \vec{y}_i^{(+)}\|^2 \text{ s. t. } \mathbf{M}^T\mathbf{M} = \mathbf{I}$$

However, this setting does not consider the complicated linguistic regularities of hypernymy relations. As seen in [10, 44], different types of hypernymy relations may correspond to different projection matrices. Denote K as the number of clusters, where each cluster corresponds to a hypernymy component. We apply K-means to $D^{(+)}$ using the vector offset $\vec{x}_i - \vec{y}_i^{(+)}$ as features. The cluster centroids are denoted as $\vec{c}_1^{(+)}, \cdots, \vec{c}_K^{(+)}$. We define $a_{i,j}^{(+)}$ as the weight of $(x_i, y_i^{(+)})$ over the dataset $D^{(+)}$ w.r.t. the jth cluster:[2]

$$a_{i,j}^{(+)} = \frac{\cos(\vec{x}_i - \vec{y}_i^{(+)}, \vec{c}_j^{(+)})}{\sum_{i'=1}^{|D^{(+)}|} \cos(\vec{x}_{i'} - \vec{y}_{i'}^{(+)}, \vec{c}_j^{(+)})}$$

Considering the orthogonal constraint, we minimize the weighted projection errors on the jth cluster with $\mathbf{M}_j^{(+)}$ being a $d \times d$ projection matrix w.r.t. the jth cluster:

$$\begin{aligned} \min J(\mathbf{M}_j^{(+)}) =& \frac{1}{2} \sum_{i=1}^{|D^{(+)}|} a_{i,j}^{(+)} \|\mathbf{M}_j^{(+)}\vec{x}_i - \vec{y}_i^{(+)}\|^2 \\ & \text{s. t. } {\mathbf{M}_j^{(+)}}^T \mathbf{M}_j^{(+)} = \mathbf{I}, \sum_{i=1}^{|D^{(+)}|} a_{i,j}^{(+)} = 1 \end{aligned} \tag{1}$$

[2] We have also experimented with several fuzzy clustering algorithms, such as Gaussian Mixture Model, Fuzzy c-Means, etc. The cluster membership probabilities are related to the weights $a_{i,j}^{(+)}$. However, they perform poorly due to the high dimensionality of word embeddings. Hence, we employ K-means to generate clusters and use a heuristic approach to compute $a_{i,j}^{(+)}$.

Let $\mathcal{M}^{(+)} = \{\mathbf{M}_1^{(+)}, \mathbf{M}_2^{(+)}, \cdots, \mathbf{M}_K^{(+)}\}$ be the set of projection parameters of all the K clusters. Learning hypernymy projections is equivalent to minimizing the following objective function:

$$\begin{aligned} \min \tilde{J}(\mathcal{M}^{(+)}) =& \frac{1}{2}\sum_{j=1}^{K}\sum_{i=1}^{|D^{(+)}|} a_{i,j}^{(+)} \|\mathbf{M}_j^{(+)}\vec{x}_i - \vec{y}_i^{(+)}\|^2 \\ & \text{s. t. } {\mathbf{M}_j^{(+)}}^T\mathbf{M}_j^{(+)} = \mathbf{I},\ \sum_{i=1}^{|D^{(+)}|} a_{i,j}^{(+)} = 1, j = 1, \ldots, K \end{aligned} \tag{2}$$

We refer to this model as the Multi-Wahba Projection (MWP) model. In mathematics, the Wahba's problem deals with three-dimensional vector observations between two coordinate systems, extensively applied in satellite attitude determination [23]. The MWP model is an extension to the Wahba's problem, considering d dimensional ($d > 3$) projections with K components. Because the values of different $\mathbf{M}_j^{(+)}$ ($j = 1, 2, \cdots, K$) are independent, the optimal solution to MWP is the same as the combination of minimizing K objectives $J(\mathbf{M}_j^{(+)})$. In this work, we pay attention to a closed-form solution to the Wahba's problem [22] based on Singular Value Decomposition (SVD) and extend it to d dimensions ($d > 3$):

Theorem 3.2. *The d-dimensional Wahba's problem has a closed-form solution as follows:*

(1) $\mathbf{B}_j = \sum_{i=1}^{|D^{(+)}|} a_{i,j}^{(+)} \vec{y}_i^{(+)} \vec{x}_i^T$;
(2) $SVD(\mathbf{B}_j) = \mathbf{U}_j \Sigma_j \mathbf{V}_j^T$;
(3) $\mathbf{R}_j = diag(\underbrace{1, \ldots, 1}_{d-1}, det(\mathbf{U}_j)det(\mathbf{V}_j))$;
(4) $\mathbf{M}_j^{(+)} = \mathbf{U}_j \mathbf{R}_j \mathbf{V}_j^T$;

Proof. For simplicity, we omit the subscript j and the superscript (+) of all variables. Eq. (1) can be re-written as:

$$J(\mathbf{M}) = \frac{1}{2}\sum_i a_i \|\mathbf{M}\vec{x}_i - \vec{y}_i\|^2 \ \text{ s. t. } \ \mathbf{M}^T\mathbf{M} = \mathbf{I}$$

Here, each pair (x_i, y_i) has the weight a_i over the entire dataset. Because $\sum_i a_i = 1$, based on the proof of the original Wahba's problem [43], we re-write the objective function as follows:

$$J(\mathbf{M}) = 1 - \sum_i a_i \vec{y}_i^T \mathbf{M}\vec{x}_i = 1 - \text{tr}(\mathbf{M}\mathbf{B}^T) \tag{3}$$

where $\mathbf{B} = \sum_i a_i \vec{y}_i \vec{x}_i^T$. The SVD of the matrix $\mathbf{B}$ is given by: $\text{SVD}(\mathbf{B}) = \mathbf{U}\Sigma\mathbf{V}^T$ with the singular value matrix $\Sigma = \text{diag}(\lambda_1, \ldots, \lambda_d)$.

Based on the result of SVD, we define two orthogonal matrices and one diagonal matrix:

$$\mathbf{U}_+ = \mathbf{U}\text{diag}(\underbrace{1, \ldots, 1}_{d-1}, \det(\mathbf{U})) \quad \mathbf{V}_+ = \mathbf{V}\text{diag}(\underbrace{1, \ldots, 1}_{d-1}, \det(\mathbf{V}))$$

$$\Sigma' = \text{diag}(\lambda_1, \ldots, \lambda_{d-1}, \lambda_d \det(\mathbf{U})\det(\mathbf{V}))$$

Due to the orthogonality of $\mathbf{U}$ and $\mathbf{V}$, we have $\det(\mathbf{U})\det(\mathbf{V}) = \pm 1$. Hence, matrix $\mathbf{B}$ can be re-decomposed as: $\mathbf{B} = \mathbf{U}_+\Sigma'\mathbf{V}_+^T$.

For simplicity, let $\mathbf{W} = \mathbf{U}_+^T\mathbf{M}\mathbf{V}_+$. Based on the cyclic invariance property of the trace, we have

$$\text{tr}(\mathbf{M}\mathbf{B}^T) = \text{tr}(\mathbf{M}\mathbf{V}_+\Sigma'\mathbf{U}_+^T) = \text{tr}(\Sigma'\mathbf{U}_+^T\mathbf{M}\mathbf{V}_+) = \text{tr}(\Sigma'\mathbf{W})$$

Therefore, we re-write Eq. (3) as: $J(\mathbf{M}) = 1 - \text{tr}(\Sigma'\mathbf{W})$.

Using the Euler axis/angle parameterization for the orthogonal matrix $\mathbf{W} = \mathcal{R}(\mathbf{e}, \phi)$, we have:

$$\begin{aligned} J(\mathbf{M}) = 1 - (\sum_{i=1}^{d-1}\lambda_i + \lambda_d \det(\mathbf{U})\det(\mathbf{V})) + (1 - \cos\phi)[\sum_{i=2}^{d-1}\lambda_i \\ + \lambda_d\det(\mathbf{U})\det(\mathbf{V}) + \sum_{i=2}^{d-1}(\lambda_1 - \lambda_i)e_i^2 + (\lambda_1 - \lambda_d\det(\mathbf{U})\det(\mathbf{V}))e_d^2] \end{aligned}$$

where $\mathbf{e} = [e_1, e_2, \cdots, e_d]$ is a unit vector and ϕ is a rotation angle. It is easy to show that $J(\mathbf{M})$ is minimized when $\cos\phi = 1$. We have:

$$\min J(\mathbf{M}) = 1 - (\sum_{i=1}^{d-1}\lambda_i + \lambda_d\det(\mathbf{U})\det(\mathbf{V})) = 1 - \text{tr}(\Sigma')$$

In this condition, we have $\mathbf{W} = \mathbf{I}$. This gives the optimal solution:

$$\mathbf{M}_{opt} = \mathbf{U}_+\mathbf{V}_+^T = \mathbf{U}\text{diag}(\underbrace{1, \ldots, 1}_{d-1}, \det(\mathbf{U})\det(\mathbf{V}))\mathbf{V}^T$$

□

Complexity analysis. For simplicity, let $n = |D^{(+)}|$. The construction of the data matrix $\mathbf{B}_j$ takes $O(nd^2)$ time. Subsequent steps include the SVD, the determinant computation and the matrix multiplication of $d \times d$ matrices. The time complexity is $O(d^3)$. Hence, the total time complexity is $O(nd^2 + d^3)$. Because d is a small constant, usually around $50 \sim 300$ in practice, and n is the size of the training set, the algorithm is sufficiently efficient for the task.

3.3 Learning Non-hypernymy Projections

In the semantic relation classification task, non-hypernyms of a term can be co-hyponyms, synonyms, etc [49]. Hence, MWP is also suitable for modeling the semantics of different non-hypernymy relations. The objective is defined as follows:[3]

$$\begin{aligned} \min \tilde{J}(\mathcal{M}^{(-)}) =& \frac{1}{2}\sum_{j=1}^{K}\sum_{i=1}^{|D^{(-)}|} a_{i,j}^{(-)} \|\mathbf{M}_j^{(-)}\vec{x}_i - \vec{y}_i^{(-)}\|^2 \\ & \text{s. t. } {\mathbf{M}_j^{(-)}}^T\mathbf{M}_j^{(-)} = \mathbf{I},\ \sum_{i=1}^{|D^{(-)}|} a_{i,j}^{(-)} = 1, j = 1, \ldots, K \end{aligned}$$

with $\vec{c}_1^{(-)}, \cdots, \vec{c}_K^{(-)}$ as non-hypernymy cluster centroids, and $\mathcal{M}^{(-)} = \{\mathbf{M}_1^{(-)}, \mathbf{M}_2^{(-)}, \cdots, \mathbf{M}_K^{(-)}\}$ as projection parameters. The centroids are generated by K-means over $D^{(-)}$, using $\vec{x}_i - \vec{y}_i^{(-)}$ as features.

3.4 Hypernymy Prediction Classifier

This part describes how MWP benefits hypernymy prediction by training a neural network classifier.

Due to the effectiveness of the Diff model for hypernymy classification [26], given a pair $(x_i, y_i) \in D^{(+)} \cup D^{(-)}$, we generate two groups of features:[4]

$$\mathcal{F}^{(+)}(\vec{x}_i, \vec{y}_i) = (\mathbf{M}_1^{(+)}\vec{x}_i - \vec{y}_i) \oplus \cdots \oplus (\mathbf{M}_K^{(+)}\vec{x}_i - \vec{y}_i)$$

$$\mathcal{F}^{(-)}(\vec{x}_i, \vec{y}_i) = (\mathbf{M}_1^{(-)}\vec{x}_i - \vec{y}_i) \oplus \cdots \oplus (\mathbf{M}_K^{(-)}\vec{x}_i - \vec{y}_i)$$

[3] Note that the numbers of clusters in hypernymy and non-hypernymy relations can also be different. For simplicity, we set the numbers of clusters in hypernymy and non-hypernymy relations uniformly as K.
[4] For simplicity, we omit the superscripts (+) and (−) for y_i here.

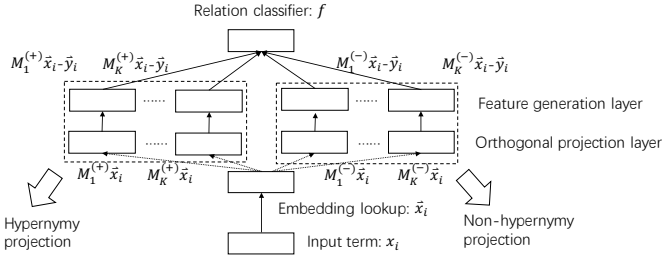


Figure 2: The MWP based neural network architecture for monolingual hypernymy prediction.

where $\oplus$ is the vector concatenation operator.

Figure 2 shows the MWP based neural network for hypernymy relation classification. For each pair $(x_i, y_i) \in D^{(+)} \cup D^{(-)}$, the model generates features $\mathcal{F}^{(+)}(\vec{x}_i, \vec{y}_i) \cup \mathcal{F}^{(-)}(\vec{x}_i, \vec{y}_i)$. A binary classifier f is trained over $D^{(+)}$ and $D^{(-)}$.[5]

The reason for designing such architecture is as follows. In distributional semantics, using embeddings $\vec{x}_i$ and $\vec{y}_i$ to predict the relation between x_i and y_i can cause the "lexical memorization" problem [18]. For example, if the classifier receives positive pairs "(dog, animal)", "(cat, animal)" and "(sheep, animal)" during training, it will "memorize" the property of the embeddings of "animal", where "animal" is regarded as a "prototypical hypernym" by [18]. In consequence, the classifier is likely to predict there is a hypernymy relation in "(rose, animal)" when seeing the term "animal".

In our approach, if there is a hypernymy relation in (x_i, y_i), the norms of the features $\mathcal{F}^{(+)}(x_i, y_i)$ are likely to be small based on Eq. (2). The norms of features $\mathcal{F}^{(-)}(x_i, y_i)$ are likely to be large. Hence the combined features $\mathcal{F}^{(+)}(x_i, y_i) \cup \mathcal{F}^{(-)}(x_i, y_i)$ are discriminative for distinguishing hypernymy vs. non-hypernymy relations. By modeling the semantics of hypernymy and non-hypernymy, this method avoids "lexical memorization". The high-level training procedure is summarized in Algorithm 1.

Algorithm 1 Monolingual Hypernymy Prediction

```
Perform K-means over D^(+), with features as x⃗_i − y⃗_i^(+);
Perform K-means over D^(−), with features as x⃗_i − y⃗_i^(−);
for j = 1 to cluster number K do
    Learn M_j^(+) and M_j^(−) by the closed-form solution;
end for
for each pair (x_i, y_i^(+)) ∈ D^(+) do
    Compute F^(+)(x⃗_i, y⃗_i^(+)) and F^(−)(x⃗_i, y⃗_i^(+));
end for
for each pair (x_i, y_i^(−)) ∈ D^(−) do
    Compute F^(+)(x⃗_i, y⃗_i^(−)) and F^(−)(x⃗_i, y⃗_i^(−));
end for
Train neural network classifier f over D^(+) and D^(−);
```

4 CROSS-LINGUAL MODELS

In this section, we introduce two projection models (i.e., TMWP and ITMWP) based on MWP for cross-lingual hypernymy prediction.

[5]We have added more hidden layers between the feature generation layer and the output layer with no improvement. Hence, an output layer is directly connected to the feature generation layer.

4.1 Transfer MWP

Let $D_S^{(+)}$, $D_S^{(-)}$, $D_T^{(+)}$ and $D_T^{(-)}$ be the training sets of hypernymy and non-hypernymy relations from the source and target languages, respectively. For low-resourced languages, the training sets are usually highly limited in size. Hence, we impose two assumptions, i.e., $|D_S^{(+)}| \gg |D_T^{(+)}|$ and $|D_S^{(-)}| \gg |D_T^{(-)}|$. We define the task of cross-lingual hypernymy prediction as follows:

Definition 4.1. (Cross-lingual Hypernymy Prediction) The goal is to train a classifier f over $D_S^{(+)}$, $D_S^{(-)}$ of the source language and $D_T^{(+)}$, $D_T^{(-)}$ of the target language, to predict hypernymy relations $U_T = \{(x_i, y_i)\}$ of the target language.

Recently, significant progress has been made in neural machine translation [39]. However, such translation technique is not sufficient to solve our task by translating training sets of the source language to the target language directly. The reasons are twofold. i) The training of neural machine translation models require a large amount of high-quality bilingual data in order to achieve high accuracy. For low-resourced languages, acquiring such data is difficult, or sometimes even infeasible [55]. ii) Cross-lingual hypernymy prediction requires the translation of words without any contexts, rather than complete sentences.

In this work, we adopt Conneau et al. [8] to learn the mappings among words across different languages. Before we train TMWP or ITMWP models, we follow the work [8] and learn a $d \times d$ projection matrix $\mathbf{S}$, which maps the embedding vector $\vec{x}$ in the source language space to $\vec{x}^*$ in the target language space such that x and x^* are terms of two languages share the same meaning.

4.1.1 Learning Hypernymy Projections. Similar to the monolingual case, the word pairs from $D_S^{(+)}$ and $D_T^{(+)}$ are grouped into K clusters. If $(x_i, y_i^{(+)}) \in D_T^{(+)}$, the features that we use for clustering are $\vec{x}_i - \vec{y}_i^{(+)}$; otherwise, we use $\mathbf{S}\vec{x}_i - \mathbf{S}\vec{y}_i^{(+)}$ as features. The objective function w.r.t. the jth cluster is as follows:

$$\begin{aligned}\min J(\mathbf{M}_j^{(+)}) =& \frac{\beta}{2} \sum_{i=1}^{|D_S^{(+)}|} a_{i,j}^{(+)} \gamma_i^{(+)} \|\mathbf{M}_j^{(+)} \mathbf{S}\vec{x}_i - \mathbf{S}\vec{y}_i^{(+)}\|^2 \\ &+ \frac{1-\beta}{2} \sum_{i=1}^{|D_T^{(+)}|} a_{i,j}^{(+)} \|\mathbf{M}_j^{(+)} \vec{x}_i - \vec{y}_i^{(+)}\|^2 \\ \text{s. t. } & \mathbf{M}_j^{(+)T} \mathbf{M}_j^{(+)} = \mathbf{I}, \sum_{i=1}^{|D_S^{(+)}|} a_{i,j}^{(+)} \gamma_i^{(+)} = 1, \sum_{i=1}^{|D_T^{(+)}|} a_{i,j}^{(+)} = 1 \end{aligned} \tag{4}$$

where $\gamma_i^{(+)}$ is a weight factor that imposes different scores to hypernymy relations in $D_S^{(+)}$. $\beta \in (0, 1)$ is a pre-defined balance factor that gives different importance to losses of the source and target languages. We define the unnormalized score $\tilde{\gamma}_i^{(+)}$ by:

$$\tilde{\gamma}_i^{(+)} = \cos(\mathbf{S}\vec{x}_i - \mathbf{S}\vec{y}_i^{(+)}, \frac{1}{|D_T^{(+)}|} \sum_{(x_j, y_j^{(+)}) \in D_T^{(+)}} \vec{x}_j - \vec{y}_j^{(+)})$$

which qualifies the semantic similarity between a pair $(x_i, y_i^{(+)}) \in D_S^{(+)}$ and all the hypernymy relations in $D_T^{(+)}$. Because $\sum_{i=1}^{|D_S^{(+)}|} a_{i,j}^{(+)} \gamma_i^{(+)} = 1$, $\gamma_i^{(+)}$ is computed by normalizing $\tilde{\gamma}_i^{(+)}$: $\gamma_i^{(+)} = \frac{\tilde{\gamma}_i^{(+)}}{\sum_{i'=1}^{|D_S^{(+)}|} a_{i',j}^{(+)} \gamma_{i'}^{(+)}}$.

After all the weights are computed, $\mathbf{M}_j^{(+)}$ can be learned by minimizing $J(\mathbf{M}_j^{(+)})$. We present the following theorem to compute the optimal solution of $\mathbf{M}_j^{(+)}$ in Eq. (4):

Theorem 4.2. *Eq. (4) is a variant of the d-dimensional Wahba's problem, which has a closed-form solution as follows:*

(1) $\mathbf{B}_j = \beta \sum_{i=1}^{|D_S^{(+)}|} a_{i,j}^{(+)} \gamma_i^{(+)} \mathbf{S}\vec{y}_i^{(+)} (\mathbf{S}\vec{x}_i)^T + (1-\beta) \sum_{i=1}^{|D_T^{(+)}|} a_{i,j}^{(+)} \vec{y}_i^{(+)} \vec{x}_i^T$;
(2) $SVD(\mathbf{B}_j) = \mathbf{U}_j \Sigma_j \mathbf{V}_j^T$;
(3) $\mathbf{R}_j = diag(\underbrace{1, \ldots, 1}_{d-1}, det(\mathbf{U}_j) det(\mathbf{V}_j))$;
(4) $\mathbf{M}_j^{(+)} = \mathbf{U}_j \mathbf{R}_j \mathbf{V}_j^T$;

Proof. We omit the subscript j and the superscript $(+)$ of all variables in the objective function. Eq. (4) can be re-written as:

$$\min J(\mathbf{M}) = \frac{\beta}{2} \sum_{i=1}^{|D_S|} a_i \gamma_i \|\mathbf{M}\mathbf{S}\vec{x}_i - \mathbf{S}\vec{y}_i\|^2 + \frac{1-\beta}{2} \sum_{i=1}^{|D_T|} a_i \|\mathbf{M}\vec{x}_i - \vec{y}_i\|^2$$
$$\text{s. t. } \mathbf{M}^T\mathbf{M} = \mathbf{I}, \sum_{i=1}^{|D_S|} a_i \gamma_i = 1, \sum_{i=1}^{|D_T|} a_i = 1$$

Hence, $\beta \sum_{i=1}^{|D_S|} a_i \gamma_i + (1-\beta) \sum_{i=1}^{|D_T|} a_i = 1$. We can see that each pair $(x_i, y_i) \in D_S$ has the weight $\beta a_i \gamma_i$, and each pair $(x_i, y_i) \in D_T$ has the weight $(1-\beta) a_i$. Define the matrix $\mathbf{B}$ as:

$$\mathbf{B} = \beta \sum_{i=1}^{|D_S|} a_i \gamma_i \mathbf{S}\vec{y}_i (\mathbf{S}\vec{x}_i)^T + (1-\beta) \sum_{i=1}^{|D_T|} a_i \vec{y}_i \vec{x}_i^T$$

The objective $J(\mathbf{M})$ in Eq. (4) can be re-written as follows: $J(\mathbf{M}) = 1 - \mathrm{tr}(\mathbf{M}\mathbf{B}^T)$. Therefore, we turn Eq. (4) into the d-dimensional Wahba's problem. The rest of the steps are the same as the previous proof, which are omitted here. □

Complexity analysis. For simplicity, let $n_1 = |D_S^{(+)}|$ and $n_2 = |D_T^{(+)}|$. The construction of $\mathbf{B}_j$ takes $O((n_1+n_2)d^2)$ time. The total time complexity is $O((n_1+n_2)d^2 + d^3)$.

Combining the objective functions w.r.t. all K clusters, the objective function for learning projected hypernymy embeddings in the cross-lingual case is derived as follows:

$$\min \tilde{J}(\mathcal{M}^{(+)}) = \frac{\beta}{2} \sum_{j=1}^{K} \sum_{i=1}^{|D_S^{(+)}|} a_{i,j}^{(+)} \gamma_i^{(+)} \|\mathbf{M}_j^{(+)} \mathbf{S}\vec{x}_i - \mathbf{S}\vec{y}_i^{(+)}\|^2$$
$$+ \frac{1-\beta}{2} \sum_{j=1}^{K} \sum_{i=1}^{|D_T^{(+)}|} a_{i,j}^{(+)} \|\mathbf{M}_j^{(+)} \vec{x}_i - \vec{y}_i^{(+)}\|^2$$
$$\text{s. t. } {\mathbf{M}_j^{(+)}}^T \mathbf{M}_j^{(+)} = \mathbf{I}, \sum_{i=1}^{|D_S^{(+)}|} a_{i,j}^{(+)} \gamma_i^{(+)} = 1, \sum_{i=1}^{|D_T^{(+)}|} a_{i,j}^{(+)} = 1,$$
$$j = 1, \ldots, K$$

which is referred as the Transfer Multi-Wahba Projection (TMWP) model.

4.1.2 Learning Non-hypernymy Projections. Similarly, to learn non-hypernymy projections, we group $D_S^{(-)}$ and $D_T^{(-)}$ into K clusters, using $\vec{x}_i - \vec{y}_i^{(-)}$ and $\mathbf{S}\vec{x}_i - \mathbf{S}\vec{y}_i^{(-)}$ as features, respectively. The objective function of TMWP is defined as follows:

$$\min \tilde{J}(\mathcal{M}^{(-)}) = \frac{\beta}{2} \sum_{j=1}^{K} \sum_{i=1}^{|D_S^{(-)}|} a_{i,j}^{(-)} \gamma_i^{(-)} \|\mathbf{M}_j^{(-)} \mathbf{S}\vec{x}_i - \mathbf{S}\vec{y}_i^{(-)}\|^2$$
$$+ \frac{1-\beta}{2} \sum_{j=1}^{K} \sum_{i=1}^{|D_T^{(-)}|} a_{i,j}^{(-)} \|\mathbf{M}_j^{(-)} \vec{x}_i - \vec{y}_i^{(-)}\|^2$$
$$\text{s. t. } {\mathbf{M}_j^{(-)}}^T \mathbf{M}_j^{(-)} = \mathbf{I}, \sum_{i=1}^{|D_S^{(-)}|} a_{i,j}^{(-)} \gamma_i^{(-)} = 1, \sum_{i=1}^{|D_T^{(-)}|} a_{i,j}^{(-)} = 1,$$
$$j = 1, \ldots, K$$

The learning process of the parameters $\mathcal{M}^{(-)}$ is the same as $\mathcal{M}^{(+)}$.

4.1.3 Hypernymy Prediction Classifier. After $\mathcal{M}^{(+)}$ and $\mathcal{M}^{(-)}$ are learned, we train the hypernymy relation classifier f for the cross-lingual case, with the procedure illustrated in Algorithm 2.

Algorithm 2 Cross-lingual Hypernymy Prediction (TMWP)

```
Perform K-means over D_S^(+) and D_T^(+), with features as S x_i − S y_i^(+) and x_i − y_i^(+), respectively;
Perform K-means over D_S^(−) and D_T^(−), with features as S x_i − S y_i^(−) and x_i − y_i^(−), respectively;
for j = 1 to cluster number K do
    Learn M_j^(+) and M_j^(−) by the closed-form solution;
end for
for each pair (x_i, y_i^(+)) ∈ D_S^(+) do
    Compute F^(+)(S x_i, S y_i^(+)) and F^(−)(S x_i, S y_i^(+));
end for
for each pair (x_i, y_i^(+)) ∈ D_T^(+) do
    Compute F^(+)(x_i, y_i^(+)) and F^(−)(x_i, y_i^(+));
end for
for each pair (x_i, y_i^(−)) ∈ D_S^(−) do
    Compute F^(+)(S x_i, S y_i^(−)) and F^(−)(S x_i, S y_i^(−));
end for
for each pair (x_i, y_i^(−)) ∈ D_T^(−) do
    Compute F^(+)(x_i, y_i^(−)) and F^(−)(x_i, y_i^(−));
end for
Train neural network classifier f over D_S^(+), D_S^(−), D_T^(+) and D_T^(−);
```

4.2 Iterative Transfer MWP

Because the sizes of $D_T^{(+)}$ and $D_T^{(-)}$ are highly limited, TMWP can encode very little knowledge from the target language. As different languages may have culture-specific words whose semantics can not be acquired using transfer learning, the performance is likely to suffer [7].

To further improve the performance, we introduce the Iterative Transfer Multi-Wahba Projection (ITMWP) model. Recall that $U_T = \{(x_i, y_i)\}$ is the collection of unlabeled term pairs of the target

Relation	BLESS	ENTAILMENT
Hypernymy	1,337	1,385
Other relations (Non-hypernymy)	13,210	1,385

Table 1: Statistic summarization of two general-domain hypernymy datasets.

language. The algorithmic procedure is shown in Algorithm 3. In each iteration, after $\mathcal{M}^{(+)}$ and $\mathcal{M}^{(-)}$ are learned, for each pair $(x_i, y_i) \in U_T$, we compute the confidence score as follows:

$$conf(x_i, y_i) = \frac{|\|\mathcal{F}^{(+)}(\vec{x}_i, \vec{y}_i)\|_2 - \|\mathcal{F}^{(-)}(\vec{x}_i, \vec{y}_i)\|_2|}{\max\{\|\mathcal{F}^{(+)}(\vec{x}_i, \vec{y}_i)\|_2, \|\mathcal{F}^{(-)}(\vec{x}_i, \vec{y}_i)\|_2\}}$$

We use the confident score instead of the output of the classifier, because modern neural networks do not generate calibrated probabilistic distributions [11]. Given a threshold τ, if $conf(x_i, y_i) > \tau$, we add the pair (x_i, y_i) to the training set (either $D_T^{(+)}$ or $D_T^{(-)}$, depending on the prediction label). The TMWP models are iteratively trained over enlarged datasets until the performance does not increase over the development set. The detailed procedure of ITMWP is shown in Algorithm 3.

Algorithm 3 Cross-lingual Hypernymy Prediction (ITMWP)

```
Train TMWP over D_S^(+), D_S^(-), D_T^(+) and D_T^(-) by Algorithm 2;
while not converge do
    for each pair (x_i, y_i) ∈ U_T do
        if conf(x_i, y_i) > τ then
            if f(x_i, y_i) = HYPERNYMY then
                Update D_T^(+) = D_T^(+) ∪ {(x_i, y_i)};
            else
                Update D_T^(-) = D_T^(-) ∪ {(x_i, y_i)};
            end if
            Update U_T = U_T \ {(x_i, y_i)}
        end if
    end for
    Update TMWP over D_S^(+), D_S^(-), D_T^(+) and D_T^(-) by Algorithm 2;
end while
```

5 MONOLINGUAL EXPERIMENTS

In this section, we conduct extensive experiments to evaluate MWP over the tasks of supervised hypernymy detection and unsupervised hypernymy relation classification. We compare it with state-of-the-art approaches to make the convincing conclusion.

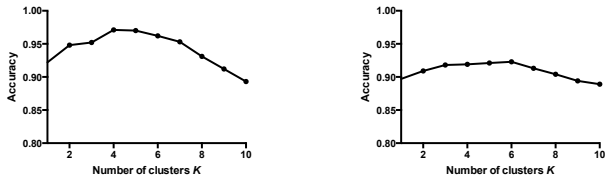
5.1 General-domain Supervised Hypernymy Detection

5.1.1 Task Description. We first evaluate MWP by supervised hypernymy detection in the general domain, which aims at classifying whether the hypernymy relation holds in a term pair. Experiments are conducted over two benchmark datasets that are frequently used in the NLP community: BLESS [2] and ENTAILMENT [1], consisting of 14,547 and 2,770 word pairs with labeled relations (i.e., hypernymy relations or other relations), respectively. The statistics of the two datasets are summarized in Table 1.

To obtain all the term embeddings for learning projection matrices and the hypernymy relation classifier, we use the same procedure as in Bojanowski et al. [4] to train fastText word embeddings over the entire Wikipedia corpus, with the dimensionality of word embeddings set to $d = 300$.

Method	BLESS	ENTAILMENT
Mikolov et al. [24]	0.84	0.83
Yu et al. [54]	0.90	0.87
Luu et al. [20]	0.93	0.91
Nguyen et al. [26]	0.94	0.91
MWP (Non-orthogonal)	**0.95**	0.90
MWP	**0.97**	**0.92**

Table 2: Performance comparison of general-domain supervised hypernymy detection in terms of accuracy.



(a) Dataset: BLESS (b) Dataset: ENTAILMENT

Figure 3: Parameter analysis of K over two datasets.

For evaluation, we follow exactly the same "leave-one-out" evaluation protocols that have been used in a series of NLP papers [20, 26, 54]. For the BLESS dataset [2], because the relations are related to 200 most frequent nouns in WordNet [25], we randomly select relations w.r.t. one noun for testing, and train the MWP model and the relation prediction classifier on others. For the ENTAILMENT dataset [1], one hypernymy relation is randomly selected for testing and the model is trained over others. All the experimental results are reported in averaged accuracy.

5.1.2 Baselines. To evaluate whether the learned features are useful to distinguishing hypernymy vs. non-hypernymy relations, we compare MWP against five hypernymy embedding methods:[6]

- Word2Vec [24]: It is the standard Skip-gram model, trained via the negative sampling technique.
- Yu et al. [54]: It learns hypernymy embeddings for terms based on the Probase taxonomy [51]. The algorithm is optimized by a max-margin neural network.
- Luu et al. [20]: It improves hypernymy embedding learning by a dynamic weighting neural model.
- HyperVec [26]: It combines the negative sampling based Skip-gram model [24] and the WordNet concept hierarchy [25] to learn hypernymy embeddings.
- MWP (Non-orthogonal): It is a variant of the MWP model, without orthogonal constraints on projection matrices.

5.1.3 Experimental Results. The experimental results are summarized in Table 2. In the implementation, as a default setting, we fix the number of clusters of MWP as $K = 4$. The parameter analysis of K will be presented further. From the results, we can see that MWP outperforms all previous baseline approaches. Simple models such as Word2Vec [24] do not produce satisfactory results because

[6]Based on the respective papers, the features of [20, 24, 54] are the hypernymy embeddings of two terms and the vector difference. The features of [26] are the vector difference and cosine similarity between two terms' embeddings, together with the magnitudes of the two embeddings. An SVM classifier is trained over all the features to make the prediction.

Statistics	ANIMAL	PLANT	VEHICLE
# Distinct terms	659	520	117
# Hypernymy relations	4,164	2,266	283
# Random word pairs	8,471	4,520	586

Table 3: Statistic summarization of three domain-specific hypernymy datasets.

Method	ANIMAL	PLANT	VEHICLE
Mikolov et al. [24]	0.80	0.81	0.82
Yu et al. [54]	0.67	0.65	0.70
Luu et al. [20]	0.89	0.92	0.89
Nguyen et al. [26]*	0.83	0.91	0.83
MWP (Non-orthogonal)	**0.90**	**0.92**	0.87
MWP	**0.92**	**0.94**	**0.90**

Table 4: Performance comparison of domain-specific supervised hypernymy detection in terms of accuracy. * refers to the result based on our own implementation.

the semantics of hypernymy relations are not explicitly modeled. Hence, they are probable to suffer from the "lexical memorization" problem. Compared to the strongest competitor HyperVec [26], the accuracy of MWP is higher by 3% over BLESS and higher by 1% over ENTAILMENT. The comparison between the MWP model and the variant MWP (Non-orthogonal) shows that the orthogonal constraint improves projection learning in MWP.

As there is no gold standard to determine the number of "latent components" in a hypernymy relation dataset, the choice of parameter K in MWP is mostly heuristic. Here, we investigate how the changes of K affect the performance of MWP. We set K from 1 to 10 and report the averaged accuracy over the two datasets. The results are shown in Figure 3. As seen, the performance is not very sensitive to K if the value of K is not extremely large or small. This is because we introduce the weights $a_{i,j}^{(+)}$ and $a_{i,j}^{(-)}$ in the model, resulting in the situations where all the projections are "fuzzy". Additionally, by using the clustering technique, the accuracy of MWP is boosted by over 5% and 2%, respectively.

5.2 Domain-specific Supervised Hypernymy Detection

5.2.1 Task Description. We further evaluate the effectiveness of MWP method over domain-specific datasets. In this set of experiments, we use three datasets derived from the following domain-specific taxonomies: ANIMAL, PLANT and VEHICLE [40]. The respective three evaluation datasets are constructed by extracting all possible taxonomic relations from taxonomies as possible samples and randomly pairing two terms that are not involved in any hypernymy relations as negative samples. We use the same datasets that have been generated and released by Luu et al. [20]. The statistics are summarized in Table 3.

For evaluation, we also follow the settings of Luu et al. [20]. Each time, we hold out relations w.r.t. one term for testing and train our model on the remaining terms. The results are also reported in averaged accuracy. Because a few terms in the three datasets are multi-word expressions (e.g., American tree, half track), we treat these terms as a whole to train the fastText word embeddings [4]. The default experimental settings and baselines are the same as in the general-domain experiments. Hence, we do not repeat the details again.

5.2.2 Experimental Results. From the experimental results in Table 4, it can be concluded that the proposed MWP model has high performance for hypernymy detection in specific domains. Specifically, the non-orthogonal version of MWP outperforms state-of-the-art methods over two domain-specific datasets (PLANT and ANIMAL) and is comparable to the strongest baseline Luu et al. [20] over the other one (VEHICLE). The full implementation of MWP outperforms all the baselines over the three datasets. Another interesting observation is that methods that use general corpora to training word embeddings (i.e., [20, 24] and ours) have relatively higher performance than methods that only consider the taxonomy data (i.e., Yu et al. [54]). Nguyen et al. [26] learn hypernymy embeddings using the general WordNet concept hierarchy, but still have relatively low performance in specific domains. This is because concepts in the taxonomy Probase [51] or in the WordNet concept hierarchy usually have low converge for specific domains, leading to low prediction performance.

5.3 Unsupervised Hypernymy Relation Classification

5.3.1 Task Description and Evaluation Protocols. This set of experiments involves two studies that compare MWP against various hypernymy measures. We follow the evaluation framework of Nguyen et al. [26], Roller et al. [30] over two benchmark datasets: BLESS and WBLESS, constructed by Kiela et al. [15] and Weeds et al. [49], respectively. In the first experiment, we aim at predicting the directionality of all the 1,337 hypernymy relations in BLESS. All the hypernymy relations are treated as positive samples and all the reverse-hypernymy relations are treated as negative samples. WBLESS is derived from BLESS, including a subset of 1,168 pairs. The experiment over WBLESS is more challenging, which is a binary classification task, aiming at distinguishing hypernymy relations and other relations (including a mixture of reverse-hypernymy, meronymy, co-hyponymy relations and randomly matched nouns).

The baseline hypernymy measures that we consider in this work are summarized as follows. Santus et al. [31],Weeds et al. [49], Kiela et al. [15] and Nguyen et al. [26] employ distributional methods for unsupervised hypernymy relation classification. Roller et al. [30] is the state-of-the art path-based measure by utilizing generalized Hearst patterns in a large text corpus.

In contrast to these hypernymy measures, the MWP model is a supervised relation classification model, instead of an unsupervised measure. Hence, MWP can not be directly applied to this task. Instead of using the MWP based classifier f, we design a hypernymy score based on the features:

$$\tilde{s}(x_i, y_i) = \|\mathcal{F}^{(-)}(\vec{x}_i, \vec{y}_i)\|_2 - \|\mathcal{F}^{(+)}(\vec{x}_i, \vec{y}_i)\|_2 \quad (5)$$

where the projection matrices w.r.t. $\mathcal{F}^{(+)}(x_i, y_i)$ are learned over hypernymy relations and the projection matrices w.r.t. $\mathcal{F}^{(-)}(x_i, y_i)$ are learned over others. For a term pair (x_i, y_i), if x_i is the hyponym and y_i is the hypernym, the norm of $\mathcal{F}^{(+)}(x_i, y_i)$ is likely to be smaller than that of $\mathcal{F}^{(-)}(x_i, y_i)$. Hence, we use $\tilde{s}(x_i, y_i) > \tilde{s}(y_i, x_i)$ to predict y_i is the hypernym of x_i.

Relation↓ Language →	fr	zh	ja	it	th	fi	el
# Hypernymy relations	4,035	2,962	1,448	3,034	1,156	7,157	2,612
# Non-hypernymy relations	8,947	6,382	3,203	6,081	1,977	9,433	1,454

Table 5: Statistics of hypernymy and non-hypernymy relation datasets of seven non-English languages. Language abbreviations: French (fr), Chinese (zh), Japanese (ja), Italian (it), Thai (th), Finnish (fn) and Greek (el).

Measure	BLESS	WBLESS
Santus et al. [31]	0.87	-
Weeds et al. [49]	-	0.75
Kiela et al. [15]	0.88	0.75
Nguyen et al. [26]	0.92	0.87
Roller et al. [30]	0.96	0.87
MWP (Non-orthogonal)	0.95	**0.89**
MWP	**0.97**	**0.92**

Table 6: Performance comparison of unsupervised hypernymy relation classification in terms of accuracy.

Because all the projection matrices require to be pre-trained before we can compute the features $\mathcal{F}^{(+)}(\vec{x}_i, \vec{y}_i)$ and $\mathcal{F}^{(-)}(\vec{x}_i, \vec{y}_i)$, we employ 14,135 hypernymy relations from the dataset Shwartz [34] as the training data, after removing pairs appearing in BLESS to avoid overfitting. Note that the proposed hypernymy measure in Eq. (5) is unsupervised because the Shwartz data is only employed to learn projection matrices. We do not train the hypernymy relation classifier to make the prediction over BLESS and WBLESS, nor do we use any BLESS or WBLESS data to learn projection matrices.

5.3.2 Experimental Results. The experimental results are summarized in Table 6. We can see that among all the baselines, Roller et al. [30] has the highest performance, with the accuracies as 96% and 87%, respectively. Compared to Roller et al. [30], the MWP model outperforms the method by 1% and 5% in terms of accuracy. Although the MWP model is primarily built for supervised hypernymy relation classification, by adding some slight modifications, it is also is highly effective for predicting the hypernymy relations in an unsupervised manner.

Note that we only focus on the binary classification of hypernymy relations in this work. In the future, we will extend our work to the multi-way classification of semantic relations, and evaluate it over other multi-way classification datasets (e.g., BIBLESS [26]).

6 CROSS-LINGUAL EXPERIMENTS

In this section, we evaluate two cross-lingual models (TMWP and ITMWP) over two tasks: cross-lingual hypernymy direction classification and cross-lingual hypernymy detection.

6.1 Datasets and Experimental Settings

We select English as the source language due to its wide usage and the availability of large training sets. We refer to the survey [45], and combine the five human-labeled datasets as the underlying training set of the source language: BLESS [2], Shwartz [34], Kotlerman [16], Turney [37] and ENTAILMENT [1]. After removing of duplicates and multi-word expressions, we create a large English dataset, consisting of 85,234 word pairs, containing 17,394 hypernymy relations and 67,930 non-hypernymy relations (including a mixture of other relations). All the datasets are available at https://chywang.github.io/data/www2019.zip.

For non-English languages, we utilize the Open Multilingual Wordnet project [5] to create training and testing sets[7]. We take seven non-English languages as target languages: French, Chinese, Japanese, Italian, Thai, Finnish and Greek. Versions of Wordnets are Wordnet Libre du Français (French), Chinese Open Wordnet, Japanese Wordnet, ItalWordnet (Italian), Thai Wordnet, FinnWordnet (Finnish) and Greek Wordnet. Hypernymy relations are generated by randomly sampling hypernymy relations from the Multilingual Wordnet concept hierarchies. Non-hypernymy relations consist of a mixture of holonymy, synonymy and randomly matched word pairs. The statistics of all the seven datasets are summarized in Table 5. For seven non-English languages, word embeddings are taken from pre-trained fastText word embeddings [4]. The cross-lingual mapping matrices are trained using the original code of Conneau et al. [8] over multi-lingual Wikipedia corpora with default parameter settings. The dimensions of word embeddings of all languages are set to $d = 300$.

6.2 Evaluation Protocols

We evaluate the two cross-lingual models (i.e., TMWP and ITMWP) over two cross-lingual hypernymy prediction tasks. The first is cross-lingual hypernymy direction classification. Its goal is to predict the directionality of hypernymy relations for the target language given the training sets in English and the target language. We take hypernymy relations as positive data, and reverse-hypernymy relations as negative data to train and evaluate our model. The second task is cross-lingual hypernymy detection. It aims at distinguishing hypernymy vs. non-hypernymy relations for the target language based on both datasets in English and the target language. In the experiments, we use all the data in English and 20% of the non-English language datasets for training, 20% for development and the rest 60% for testing, partitioned randomly. By rotating the 5-fold subsets of the non-English language datasets, we report the performance of all the models in terms of averaged accuracy.

Because pattern-based methods and a few distributional methods for hypernymy prediction is highly language dependent, they are not suitable for cross-lingual hypernymy prediction over arbitrary languages. Hence, we follow the evaluation protocols that used in Shwartz et al. [34]. We employ several state-of-the-art distributional approaches as baselines, introduced as follows:

- Santus et al. [31]: It is an entropy-based hypernymy measure SLQS that characterizes the semantic generality of terms.
- Kiela et al. [15]: It is a distributional generality measure that models the hierarchical property of hypernymy relations.
- Weeds et al. [49]: It is a supervised distributional model based on vector offsets of term pairs.
- Shwartz et al. [34]: It is a neural network architecture for relation classification. Because several non-English languages

[7] http://compling.hss.ntu.edu.sg/omw/

Method	fr	zh	ja	it	th	fi	el
Task: cross-lingual hypernymy direction classification							
Santus et al. [31]	0.65	0.65	0.68	0.61	0.63	0.70	0.62
Weeds et al. [49]	0.76	0.71	0.77	0.76	0.72	0.77	0.70
Kiela et al. [15]	0.67	0.65	0.71	0.68	0.65	0.70	0.62
Shwartz et al. [34]	0.79	0.67	0.71	0.72	0.66	0.75	0.66
TMWP (N)	0.78	0.71	0.75	0.76	0.73	0.76	0.71
TMWP	0.80	0.72	0.76	0.78	0.75	0.78	0.73
ITMWP (N)	**0.82**	0.72	0.76	0.78	0.75	**0.81**	0.72
ITMWP	0.81	**0.74**	**0.78**	**0.81**	**0.78**	**0.81**	**0.75**
Task: cross-lingual hypernymy detection							
Santus et al. [31]	0.67	0.63	0.67	0.62	0.64	0.62	0.64
Weeds et al. [49]	0.74	0.66	0.68	0.71	0.62	0.68	0.69
Kiela et al. [15]	0.70	0.61	0.65	0.68	0.57	0.61	0.67
Shwartz et al. [34]	0.72	0.66	0.69	0.64	0.66	0.69	0.70
TMWP (N)	0.72	0.67	0.70	0.70	0.68	0.71	0.70
TMWP	0.75	0.71	0.76	0.72	0.69	0.72	0.71
ITMWP (N)	0.72	**0.74**	0.77	**0.74**	0.67	0.71	0.72
ITMWP	**0.76**	0.73	**0.78**	**0.74**	**0.72**	**0.73**	**0.73**

Table 7: Performance comparison of two cross-lingual hypernymy prediction tasks in terms of accuracy. TMWP (N) and ITMWP (N) are stand for TMWP (Non-orthogonal) and ITMWP (Non-orthogonal).

(e.g., Chinese, Thai) lack high-quality hypernymy patterns. The path-based sub-networks are not implemented.

- TMWP (Non-orthogonal) and ITMWP (Non-orthogonal): They are the variants of TMWP and ITMWP without the orthogonal constraint on projection matrices, respectively.

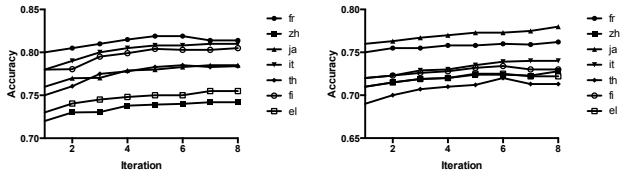
Apart from the proposed two models and their variants, the remaining competing approaches [15, 31, 34, 49] are not designed for cross-lingual hypernymy prediction. To implement all the baselines for the two cross-lingual hypernymy prediction tasks, we employ Conneau et al. [8] to translate the embeddings of terms in the source language to those of the target languages. Next, we train all the baseline models over the English dataset after translating it to the target language and the training set of the target language. For simplicity, we set $K = 4$ and $\beta = 0.5$ for TMWP and ITMWP and their variants in all the cross-lingual experiments.

6.3 Experimental Results

The experimental results of both tasks over all the seven non-English languages are summarized in Table 7.

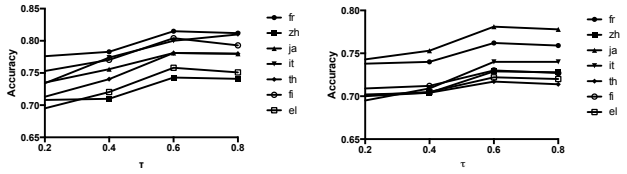
From the results, we can draw the following conclusions. i) For both tasks, ITMWP and TMWP outperform all the previous methods over all seven non-English languages. Overall, the proposed models boost the accuracy by $2\% \sim 9\%$ for cross-lingual hypernymy direction classification, depending on different languages. The performance of our models for cross-lingual hypernymy detection is similar to that of the direction classification task. ii) By using unsupervised neural word translation and the orthogonal constraint, knowledge from the source languages can be employed to improve the performance of hypernymy prediction of the target language. iii) The performance of cross-lingual hypernymy direction classification is generally higher than cross-lingual hypernymy detection, indicating the latter task is more challenging.

We further investigate how ITMWP improves the performance. We fix $\tau = 0.7$ and run our algorithm for 8 iterations over the two tasks. The results are illustrated in Figure 4. As seen, the accuracy increases steadily during the first few iterations. The performance becomes relatively stable because there are no sufficient number of relations with high confidence that can be added to training sets.



(a) Task: cross-lingual hypernymy direction classification (b) Task: cross-lingual hypernymy detection

Figure 4: Performance of ITMWP over two cross-lingual hypernymy prediction tasks.



(a) Task: cross-lingual hypernymy direction classification (b) Task: cross-lingual hypernymy detection

Figure 5: Parameter analysis of τ over two cross-lingual hypernymy prediction tasks.

Additionally, we tune the parameter τ in ITMWP and run the algorithm in 5 iterations. The results are summarized in Table 5. It shows that the changes of performance trend are similar across different languages. The value of τ reflects the trade-off between the number of pairs to be added to the training set and the accuracies of these pairs. When τ is small, the algorithm tends to add more pairs to the training set, unavoidably introducing errors to the training set. In contrast, when τ is large, the effect of training data augmentation is reduced. We suggest that the algorithm achieves the best performance when τ is set around 0.7.

7 CONCLUSION AND FUTURE WORK

In this paper, we propose a family of fuzzy orthogonal projection models for monolingual and cross-lingual hypernymy prediction. It includes three models: MWP, TMWP and ITMWP. MWP distinguishes hypernymy vs. non-hypernymy relations based on distributional fuzzy mappings from embeddings of a term to those of its hypernyms and non-hypernyms. TMWP and ITMWP are designed to transfer the semantic knowledge from the source language to target languages for cross-lingual hypernymy prediction. Experiments illustrate the effectiveness of our models over both monolingual and cross-lingual hypernymy prediction. In the future, we plan to i) extend our method to predict multiple types of semantic relations over multiple languages, and to ii) improve cross-lingual hypernymy prediction via multi-lingual embeddings.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000904.

REFERENCES

[1] 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*. 23–32.

[2] Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the Gems 2011 Workshop on Geometrical MODELS of Natural Language Semantics*. 1–10.

[3] Chris Biemann, Dmitry Ustalov, Alexander Panchenko, and Nikolay Arefyev. 2017. Negative Sampling Improves Hypernymy Extraction Based on Projection Learning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. 543–550.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *TACL* 5 (2017), 135–146.

[5] Francis Bond and Ryan Foster. 2013. Linking and Extending an Open Multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. 1352–1362.

[6] Haw-Shiuan Chang, ZiYun Wang, Luke Vilnis, and Andrew McCallum. 2018. Distributional Inclusion Vector Embedding for Unsupervised Hypernymy Detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 485–495.

[7] Nurendra Choudhary, Rajat Singh, and Manish Shrivastava. 2018. Cross-Lingual Task-Specific Representation Learning for Text Classification in Resource Poor Languages. *CoRR* abs/1806.03590 (2018). http://arxiv.org/abs/1806.03590

[8] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word Translation Without Parallel Data. *CoRR* abs/1710.04087 (2017). arXiv:1710.04087 http://arxiv.org/abs/1710.04087

[9] Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word Embeddings, Analogies, and Machine Learning: Beyond king - man + woman = queen. In *Proceedings of the 26th International Conference on Computational Linguistics*. 3519–3530.

[10] Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning Semantic Hierarchies via Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. 1199–1209.

[11] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*. 1321–1330.

[12] Amit Gupta, Francesco Piccinno, Mikhail Kozhevnikov, Marius Pasca, and Daniele Pighin. 2016. Revisiting Taxonomy Induction over Wikipedia. In *Proceedings of the 26th International Conference on Computational Linguistics*. 2300–2309.

[13] Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*. 539–545.

[14] Matthias Keller, Patrick Mühlschlegel, and Hannes Hartenstein. 2013. Search result presentation: supporting post-search navigation by integration of taxonomy data. In *Proceedings of the 22nd International World Wide Web Conference, Companion Volume*. 1269–1274.

[15] 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 and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*. 119–124.

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

[17] Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*. 75–79.

[18] Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do Supervised Distributional Methods Really Learn Lexical Inference Relations?. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 970–976.

[19] Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. 2012. Automatic taxonomy construction from keywords. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1433–1441.

[20] Anh Tuan Luu, Yi Tay, Siu Cheung Hui, and See-Kiong Ng. 2016. Learning Term Embeddings for Taxonomic Relation Identification Using Dynamic Weighting Neural Network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 403–413.

[21] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *Proceedings of the Seventh Biennial Conference on Innovative Data Systems Research*.

[22] F Landis Markley. 1988. Attitude determination using vector observations and the singular value decomposition. *Journal of the Astronautical Sciences* 36, 3 (1988), 245–258.

[23] F Landis Markley and John L Crassidis. 2014. *Fundamentals of spacecraft attitude determination and control*. Vol. 33. Springer.

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). arXiv:1301.3781 http://arxiv.org/abs/1301.3781

[25] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[26] Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical Embeddings for Hypernymy Detection and Directionality. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 233–243.

[27] Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a Large-Scale Taxonomy from Wikipedia. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*. 1440–1445.

[28] Stephen Roller and Katrin Erk. 2016. Relations such as Hypernymy: Identifying and Exploiting Hearst Patterns in Distributional Vectors for Lexical Entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2163–2172.

[29] Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet Selective: Supervised Distributional Hypernymy Detection. In *Proceedings of the 25th International Conference on Computational Linguistics*. 1025–1036.

[30] Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 358–363.

[31] Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing Hypernyms in Vector Spaces with Entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. 38–42.

[32] Enrico Santus, Vered Shwartz, and Dominik Schlechtweg. 2017. Hypernyms under Siege: Linguistically-motivated Artillery for Hypernymy Detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. 65–75.

[33] Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T. Vanni, Brian M. Sadler, and Jiawei Han. 2018. HiExpan: Task-Guided Taxonomy Construction by Hierarchical Tree Expansion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2180–2189.

[34] Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 2389–2398.

[35] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning Syntactic Patterns for Automatic Hypernym Discovery. In *Advances in Neural Information Processing Systems 17*. 1297–1304.

[36] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*. 697–706.

[37] Peter D. Turney and Saif M. Mohammad. 2015. Experiments with three approaches to recognizing lexical entailment. *Natural Language Engineering* 21, 3 (2015), 437–476.

[38] Shyam Upadhyay, Yogarshi Vyas, Marine Carpuat, and Dan Roth. 2018. Robust Cross-Lingual Hypernymy Detection Using Dependency Context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 607–618.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 6000–6010.

[40] Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn Reloaded: A Graph-Based Algorithm for Taxonomy Induction. *Computational Linguistics* 39, 3 (2013), 665–707.

[41] Ivan Vulic, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. HyperLex: A Large-Scale Evaluation of Graded Lexical Entailment. *Computational Linguistics* 43, 4 (2017).

[42] Yogarshi Vyas and Marine Carpuat. 2016. Sparse Bilingual Word Representations for Cross-lingual Lexical Entailment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1187–1197.

[43] Grace Wahba. 1965. A least squares estimate of satellite attitude. *SIAM review* 7, 3 (1965), 409–409.

[44] Chengyu Wang and Xiaofeng He. 2016. Chinese Hypernym-Hyponym Extraction from User Generated Categories. In *Proceedings of the 26th International Conference on Computational Linguistics*. 1350–1361.

[45] Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2017. A Short Survey on Taxonomy Learning from Text Corpora: Issues, Resources and Recent Advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1190–1203.

[46] Chengyu Wang, Junchi Yan, Aoying Zhou, and Xiaofeng He. 2017. Transductive Non-linear Learning for Chinese Hypernym Prediction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 1394–1404.

[47] Zhigang Wang, Juanzi Li, Shuangjie Li, Mingyang Li, Jie Tang, Kuo Zhang, and Kun Zhang. 2014. Cross-Lingual Knowledge Validation Based Taxonomy Derivation from Heterogeneous Online Wikis. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. 180–186.

[48] Zhongyuan Wang, Kejun Zhao, Haixun Wang, Xiaofeng Meng, and Ji-Rong Wen. 2015. Query Understanding through Knowledge-Based Conceptualization. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. 3264–3270.

[49] Julie Weeds, Daoud Clarke, Jeremy Reffin, David J. Weir, and Bill Keller. 2014. Learning to Distinguish Hypernyms and Co-Hyponyms. In *Proceedings of the 25th International Conference on Computational Linguistics*. 2249–2259.

[50] Tianxing Wu, Lei Zhang, Guilin Qi, Xuan Cui, and Kang Xu. 2017. Encoding Category Correlations into Bilingual Topic Modeling for Cross-Lingual Taxonomy Alignment. In *Proceedings of the 16th International Semantic Web Conference*. 728–744.

[51] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. 2012. Probase: a probabilistic taxonomy for text understanding. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 481–492.

[52] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation. In *Proccedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1006–1011.

[53] Josuke Yamane, Tomoya Takatani, Hitoshi Yamada, Makoto Miwa, and Yutaka Sasaki. 2016. Distributional Hypernym Generation by Jointly Learning Clusters and Projections. In *Proceedings of the 26th International Conference on Computational Linguistics*. 1871–1879.

[54] Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning Term Embeddings for Hypernymy Identification. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. 1390–1397.

[55] Poorya Zaremoodi, Wray L. Buntine, and Gholamreza Haffari. 2018. Adaptive Knowledge Sharing in Multi-Task Learning: Improving Low-Resource Neural Machine Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 656–661.

[56] Yuchen Zhang, Amr Ahmed, Vanja Josifovski, and Alexander J. Smola. 2014. Taxonomy discovery for personalized recommendation. In *Proceedings of the Seventh ACM International Conference on Web Search and Data Mining*. 243–252.