# Joint Multilingual Supervision for Cross-lingual Entity Linking

**Shyam Upadhyay**
University of Pennsylvania
Philadelphia, PA
shyamupa@seas.upenn.edu

**Nitish Gupta**
University of Pennsylvania
Philadelphia, PA
nitishg@seas.upenn.edu

**Dan Roth**
University of Pennsylvania
Philadelphia, PA
danroth@seas.upenn.edu

## Abstract

Cross-lingual Entity Linking (XEL) aims to ground entity mentions written in *any* language to an English Knowledge Base (KB), such as Wikipedia. XEL for most languages is challenging, owing to limited availability of resources as supervision. We address this challenge by developing the first XEL approach that combines supervision from multiple languages *jointly*. This enables our approach to: **(a)** augment the limited supervision in the target language with additional supervision from a high-resource language (like English), and **(b)** train a *single* entity linking model for multiple languages, improving upon individually trained models for each language. Extensive evaluation on three benchmark datasets across 8 languages shows that our approach significantly improves over the current state-of-the-art. We also provide analyses in two limited resource settings: **(a)** *zero-shot setting*, when no supervision in the target language is available, and in **(b)** *low-resource setting*, when some supervision in the target language is available. Our analysis provides insights into the limitations of zero-shot XEL approaches in realistic scenarios, and shows the value of joint supervision in low-resource settings.[1]

## 1 Introduction

Entity Linking (EL) systems ground entity mentions in text to entries in Knowledge Bases (KB), such as Wikipedia (Mihalcea and Csomai, 2007). Recently, the task of Cross-lingual Entity Linking (XEL) has gained attention (McNamee et al., 2011; Ji et al., 2015; Tsai and Roth, 2016) with the goal of grounding entity mentions written in *any* language to the English Wikipedia. For instance, Figure 1 shows a Tamil (a language with >70 million speakers) and an English mention (shown [**enclosed**]) and their mention contexts. XEL involves grounding the Tamil mention (which translates to 'Liverpool') to the football club Liverpool_F.C., and not the city or the university. XEL enables knowledge acquisition directly from documents in any language, without resorting to machine translation.

சுரேஸ் **[லிவர்பூல்]** மற்றும் உருகுவே விளையாடுகிறார்.

Everton won against **[Liverpool]** in an FA Cup match.

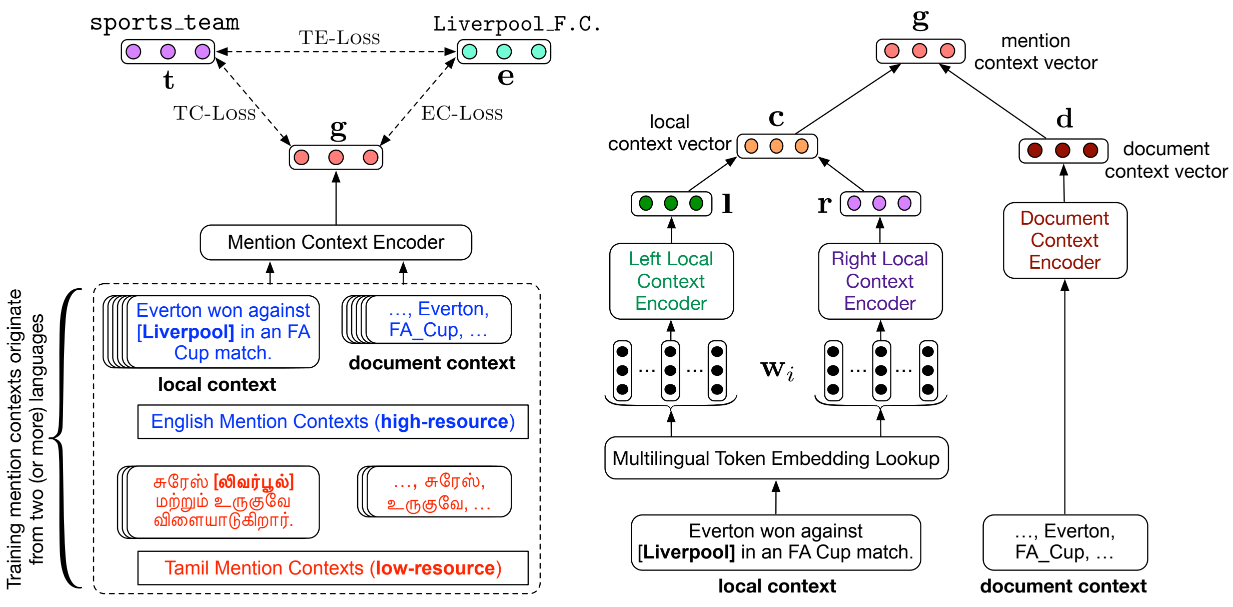
Figure 1: Tamil and English mention contexts containing [**mentions**] of the entity Liverpool_F.C. from the respective Wikipedias. Tamil Wikipedia only has 9 mentions referring to Liverpool_F.C., whereas English Wikipedia has 5303 such mentions. Clearly, there is a need to augment the limited contextual evidence in low-resource languages with evidence from high-resource languages like English. Tamil sentence translates to "Suarez plays for [**Liverpool**] and Uruguay."

Training an EL model requires grounded mentions, i.e. mentions of entities that are grounded to a Knowledge Base (KB), as supervision (Figure 1). While millions of such mentions are available in English, by virtue of hyperlinks in the English Wikipedia, this is not the case for most languages. This makes learning XEL models challenging, especially for languages with limited resources (e.g., the Tamil Wikipedia is only 1% of the English Wikipedia in size). To overcome this challenge, it is desirable to augment the limited contextual evidence available in the target language with evidence from high-resource languages like English.

We propose XELMS (XEL with Multilingual Supervision) (§2), the first approach that fulfills the above desiderata by using multilingual supervision to train an XEL model. XELMS represents the mention contexts of the same entity from different languages in the same semantic space using a single context encoder (§2.1). Language-agnostic entity representations are jointly learned with the relevant mention context representations, so that an entity and its context share similar representations.

[1] Code at www.github.com/shyamupa/xelms

*Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2495
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics


sports_team
TE-Loss
Liverpool_F.C.
t
e
TC-Loss
EC-Loss
g
Mention Context Encoder
Training mention contexts originate from two (or more) languages
Everton won against [Liverpool] in an FA Cup match.
local context
..., Everton, FA_Cup, ...
document context
English Mention Contexts (high-resource)
சுரேஸ் [லிவர்பூல்] மற்றும் உருகுவே விளையாடுகிறார்.
..., சுரேஸ், உருகுவே, ...
Tamil Mention Contexts (low-resource)
g
mention context vector
c
local context vector
d
document context vector
l
r
Left Local Context Encoder
Right Local Context Encoder
Document Context Encoder
$\mathbf{w}_i$
Multilingual Token Embedding Lookup
Everton won against [Liverpool] in an FA Cup match.
local context
..., Everton, FA_Cup, ...
document context

(a) Overview of XELMS. Mentions are shown [**enclosed**].

(b) Mention Context Encoder.

Figure 2: (**a**) Grounded mentions from two or more languages (English and Tamil shown) can be used to supervise XELMS. The context $\mathbf{g}$, entity $\mathbf{e}$ and type $\mathbf{t}$ vectors interact through Entity-Context loss (EC-LOSS), Type-Context loss (TC-LOSS) and Type-Entity loss (TE-LOSS). The Tamil sentence is the same as in Figure 1, and other mentions in it translate to [**Suarez**] and [**Uruguay**]. (**b**) The Mention Context Encoder (§2.1) encodes the local context (neighboring words) and the document context (surfaces of other mentions in the document) of the mention into $\mathbf{g}$. Internal view of local context encoder is in Figure 3.

Additionally, by encoding freely available structured knowledge, like fine-grained entity types, the entity and context representations can be further improved (§2.2).

The ability to use multilingual supervision enables XELMS to learn XEL models for target languages with limited resources by exploiting freely available supervision from high resource languages (like English). We show that XELMS outperforms existing state-of-the-art approaches that only use target language supervision, across 3 benchmark datasets in 8 languages (§5.1). Moreover, while previous XEL models (McNamee et al., 2011; Tsai and Roth, 2016) train separate models for different languages, XELMS can train a *single* model for performing XEL in multiple languages (§5.2).

One of the goals of XEL is to enable understanding of languages with limited resources. We provide experimental analyses in two such settings. In the *zero-shot setting* (§6.1), where *no* supervision is available in the target language, we show that the good performance of zero-shot XEL approaches (Sil et al., 2018) can be attributed to the use of prior probabilities. These probabilities are computed from large amount of grounded mentions, which are not available in realistic zero-shot settings. In the *low-resource setting* (§6.2), where some supervision is available in the target language, we show that even when only a fraction of the available supervision in the target language is provided, XELMS can achieve competitive performance by exploiting supervision from English.

The contributions of our work are,

- A new XEL approach, XELMS, that learns a XEL model for a language with limited resources by exploiting additional supervision from a high-resource language like English.
- XELMS can also train a *single* XEL model for multiple languages jointly, which we show improves on separately trained models.
- Analysis of XEL approaches in the zero-shot and low-resource settings. Our analysis reveals that in realistic scenarios, zero-shot XEL is not as effective as previously shown. We also show that in low-resource settings jointly training with English leads to better utilization of target language supervision.

## 2 Cross-lingual EL with XELMS

Given a mention $m$ in a document $\mathcal{D}$ written in any language, XEL involves linking $m$ to its gold entity $e^*$ in a KB, $\mathcal{K} = \{e_1, \cdots, e_n\}$.

An overview of XELMS is shown in Figure 2a. XELMS computes the probability, $\mathrm{P}_{\text{context}}(e \mid m)$, of a mention $m$ referring to entity $e \in \mathcal{K}$ using a mention context vector $\mathbf{g} \in \mathbb{R}^h$ representing

$m$'s context, and an entity vector $\mathbf{e} \in \mathbb{R}^h$, representing the entity $e \in \mathcal{K}$ (one vector per entity). XELMS can also incorporate structured knowledge like fine-grained entity types (§2.2) using a multi-task learning approach (Caruana, 1998), by learning a type vector $\mathbf{t} \in \mathbb{R}^h$ for each possible type $t$ (e.g., sports_team) associated with the entity $e$. The entity vector $\mathbf{e}$, context vector $\mathbf{g}$ and the type vector $\mathbf{t}$ are jointly trained, and interact through appropriately defined pairwise loss terms – an Entity-Context loss (EC-LOSS), Type-Entity loss (TE-LOSS) and a Type-Context loss (TC-LOSS).

The mention context vector $\mathbf{g}$ is generated by a mention context encoder (§2.1), shown in Figure 2b. The *mention context* of $m$ in a document $\mathcal{D}$ consists of: **(a)** neighboring words around the mention, which we refer to as its *local context* and, **(b)** surfaces of other mentions appearing in $\mathcal{D}$, which we refer as its *document context*.

XELMS is trained using grounded mentions in multiple languages (English and Tamil in Figure 2a), which can be derived from Wikipedia (§4.1).

### 2.1 Mention Context Representation

To learn from mention contexts in multiple languages, we generate mention context representations using a language-agnostic mention context encoder. An overview of the mention context encoder is shown in Figure 2b. Below we describe the components of the mention context encoder, namely multilingual word embeddings and local and document context encoders.

**Multilingual Word Embeddings** (Ammar et al., 2016b; Smith et al., 2017; Duong et al., 2017) jointly encode words in multiple ($\geq$2) languages in the same vector space such that semantically similar words in the same language, and translationally equivalent words in different languages are close (per cosine similarity). Multilingual embeddings generalize bilingual embeddings, which do the same for two languages *only*.

We use FASTTEXT (Bojanowski et al., 2017; Smith et al., 2017), which aligns monolingual embeddings of multiple languages in the same space using a small dictionary ($\sim$2500 pairs) from each language to English. Both monolingual embeddings and the dictionary can be easily obtained for languages with limited resources. We denote the multilingual word embeddings for a set of tokens $\{w_1, w_2, \cdots, w_n\}$ by $\mathbf{w}_{1:n} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n\}$, where each $\mathbf{w}_i \in \mathbb{R}^d$.

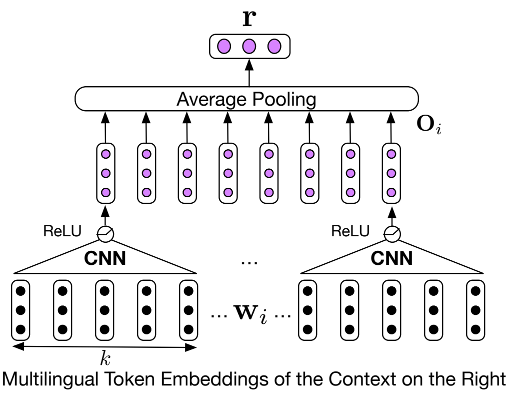


Figure 3: Local Context Encoder, for the right context. Figure 2b shows how it fits inside Mention Context Encoder.

**Local Context Representation** The local context of a mention $m$, spanning tokens $i$ to $j$, consists of left context (tokens $i - W$ to $j$) and right context (tokens $i$ to $j + W$). For example, for the mention [**Liverpool**] in Figure 2b, the left and right contexts are "Everton won against Liverpool" and "Liverpool in a FA Cup match" respectively. The local context encoder (Figure 3) encodes the left and the right contexts into vectors $\mathbf{l} \in \mathbb{R}^h$ and $\mathbf{r} \in \mathbb{R}^h$ using a convolutional neural network (CNN). These two vectors are then combined to generate the local context vector $\mathbf{c} \in \mathbb{R}^h$ (Figure 2b).

The CNN convolves continuous spans of $k$ tokens using a filter matrix $\mathbf{F} \in \mathbb{R}^{kd \times h}$ to project the concatenation ($\oplus$ operator) of the token embeddings in the span. The resulting vector is passed through a ReLU unit to generate convolutional output $\mathbf{O}_i$. The outputs $\{\mathbf{O}_i\}$ are pooled by averaging,

$$\mathbf{O}_i = \text{RELU}(\mathbf{F}^T(\mathbf{w}_i \oplus \cdots \oplus \mathbf{w}_{i+k-1})) \tag{1}$$

$$\text{ENC}(\mathbf{w}_{1:n}) = \text{AVG}(\mathbf{O}_1, \cdots, \mathbf{O}_{n-k+1}) \tag{2}$$

Left and right context vectors $\mathbf{l}$ and $\mathbf{r}$ are computed using respective $\text{ENC}(.)$ layers,

$$\mathbf{l} = \text{ENC}_{\text{left}}(\mathbf{w}_{i-W} \cdots \mathbf{w}_j) \tag{3}$$

$$\mathbf{r} = \text{ENC}_{\text{right}}(\mathbf{w}_i \cdots \mathbf{w}_{j+W}) \tag{4}$$

These vectors together generate the local context vector $\mathbf{c} = \mathcal{F}_{2h,h}(\mathbf{l} \oplus \mathbf{r})$. Here $\mathcal{F}_{d_i,d_o} : \mathbf{v}_i \rightarrow \mathbf{v}_o$ denotes a feed-forward layer that takes $\mathbf{v}_i \in \mathbb{R}^{d_i}$ as input, and outputs $\mathbf{v}_o \in \mathbb{R}^{d_o}$.

**Document Context Representation** Presence of certain mentions in a document can help disambiguate other mentions. For example, "Suarez", "Everton" in a document can help disambiguate "Liverpool". To incorporate this, we define the

document context $d_m$ of a mention $m$ appearing in document $\mathcal{D}$ to be the bag of all other mentions in $\mathcal{D}$. We encode $d_m$ into a dense document context vector $\mathbf{d} \in \mathbb{R}^h$ by a feed-forward layer $\mathbf{d} = \mathcal{F}_{|V|,h}(d_m)$. Here $V$ is the set containing all mention surfaces seen during training. When training jointly over multiple languages, $V$ consists of mention surfaces seen in all languages (e.g. all English and Tamil mention surfaces) during training. This enables parameter sharing by embedding mention surfaces in different languages in the same low-dimensional space.
The local and document context vectors $\mathbf{c}$ and $\mathbf{d}$ are combined to get the mention context vector $\mathbf{g} = \mathcal{F}_{2h,h}(\mathbf{c} \oplus \mathbf{d})$.

**Context Conditional Probability** We compute the probability of a mention $m$ linking to entity $e$ using its context vector $\mathbf{g}$ and the entity vector $\mathbf{e}$,

$$\mathrm{P}_{\text{context}}(e \mid m) = \frac{\exp(\mathbf{g}^T \mathbf{e})}{\sum\limits_{e' \in C(m)} \exp(\mathbf{g}^T \mathbf{e}')} \qquad (5)$$

where $C(m)$ denotes all candidate entities of the mention $m$ (§3.1 explains how $C(m)$ is generated). We minimize the negative log-likelihood of $\mathrm{P}_{\text{context}}(e \mid m)$ with respect to the gold entity $e^*$ against the candidate entities $C(m)$, and call it the Entity-Context loss (EC-LOSS),

$$\text{EC-LOSS} = -\log \frac{\mathrm{P}_{\text{context}}(e^* \mid m)}{\sum\limits_{e' \in C(m)} \mathrm{P}_{\text{context}}(e' \mid m)} \qquad (6)$$

### 2.2 Including Type Information

Incorporating the fine-grained types of a mention $m$ can help rank entities of the appropriate type higher than others (Ling et al., 2015; Gupta et al., 2017; Raiman and Raiman, 2018). For instance, knowing the correct type of mention **[Liverpool]** as sports_team and constraining linking to entities with the relevant type, encourages disambiguation to the correct entity.

To make the mention context representation $\mathbf{g}$ type-aware, we predict the set of fine-grained types of $m$, $\mathbf{T}(m) = \{t_1, ..., t_{|\mathbf{T}(m)|}\}$ using $\mathbf{g}$. Each $t_i$ belongs to a pre-defined type vocabulary $\Gamma$.[2] The probability of a type $t$ belonging to $\mathbf{T}(m)$ given the mention context is defined as $\mathrm{P}(t \mid m) = \sigma(\mathbf{t}^T \mathbf{g})$, where $\sigma$ is the sigmoid function and $\mathbf{t}$ is the learnable embedding for type $t$.

We define a Type-Context loss (TC-LOSS) as,

$$\text{TC-LOSS} = \mathrm{BCE}(\mathbf{T}(m), \mathrm{P}(t \mid m)) \qquad (7)$$

where BCE is the Binary Cross-Entropy Loss,

$$-\sum_{t \in \mathbf{T}(m)} \log \mathrm{P}(t \mid m) - \sum_{t \notin \mathbf{T}(m)} \log(1 - \mathrm{P}(t \mid m))$$

We also incorporate the entity-type information in the entity representations, and define a similar Type-Entity loss (TE-LOSS).

To identify the gold types $\mathbf{T}(m)$ of a mention $m$, we make the distant supervision assumption (same as Ling et al. (2015)) and assign the types of the gold entity $e^*$ to be the types of the mention. Gold fine-grained types of the entities can be acquired from resources like Freebase (Bollacker et al., 2008) or YAGO (Hoffart et al., 2013).

## 3 Training and Inference

We explain how XELMS generates candidate entities, performs inference, and combines the different training losses.

### 3.1 Candidate Generation

Candidate generation identifies a small number of plausible entities for a mention $m$ to avoid brute force comparison with all KB entities. Given $m$, candidate generation outputs a list of candidate entities $C(m) = \{e_1, e_2, \cdots, e_K\}$ of size at most $K$ (we use $K$=20), each associated with a prior probability $\mathrm{P}_{\text{prior}}(e_i \mid m)$ indicating the probability of $m$ referring to $e_i$, given only $m$'s surface. $\mathrm{P}_{\text{prior}}$ is estimated from counts over the training mentions.

We adopt Tsai and Roth (2016)'s candidate generation strategy with some minor modifications (Appendix A). Using other approaches like CrossWikis (Spitkovsky and Chang, 2012), lead to consistently worse recall. We note that transliteration based candidate generation (McNamee et al., 2011; Pan et al., 2017; Tsai and Roth, 2018; Upadhyay et al., 2018) can further improve recall.

### 3.2 Inference

We combine the context conditional entity probability $\mathrm{P}_{\text{context}}(e \mid m)$ (eq. 5) and prior probability $\mathrm{P}_{\text{prior}}(e \mid m)$ by taking their union:

$$\begin{aligned}\mathrm{P}_{\text{model}}(e \mid m) &= \mathrm{P}_{\text{prior}}(e \mid m) + \mathrm{P}_{\text{context}}(e \mid m) \\ &\quad - \mathrm{P}_{\text{prior}}(e \mid m) \times \mathrm{P}_{\text{context}}(e \mid m)\end{aligned}$$

Inference for the mention $m$ picks the entity,

$$\hat{e} = \arg\max_{e \in C(m)} \mathrm{P}_{\text{model}}(e \mid m) \qquad (8)$$

[2] We use the type vocabulary $\Gamma$ from Ling and Weld (2012), which contains 112 fine-grained types ($|\Gamma| = 112$)

### 3.3 Training Objective

When only training the mention context encoder and entity vectors, we minimize the EC-LOSS averaged over all training mentions. When using the two type-aware losses, we minimize a weighted sum of EC-LOSS, TE-LOSS, and TC-LOSS, using the weighing scheme of Kendall et al. (2018),

$$\frac{\text{EC-LOSS}}{2\lambda_{\text{EC}}^2} + \frac{\text{TE-LOSS}}{2\lambda_{\text{TE}}^2} + \frac{\text{TC-LOSS}}{2\lambda_{\text{TC}}^2} + \log \lambda_{\text{EC}}^2 + \log \lambda_{\text{TE}}^2 + \log \lambda_{\text{TC}}^2 \quad (9)$$

Here $\lambda_i$ are learnable scalar weighing parameters, and the respective $\frac{1}{2\lambda_i^2}$ and $\log \lambda_i^2$ term ensure that $\lambda_i^2$ does not grow unboundedly. This way, the model learns the relative weight for each loss term.

During training, mentions from different languages are mixed using *inverse-ratio mini-batch mixing* strategy. That is, if two languages have training data sizes proportional to $\alpha : \beta$, at any time during training, mini-batches seen from them are in the ratio $\frac{1}{\alpha} : \frac{1}{\beta}$. This strategy prevents languages with more training data from overwhelming languages with less training data. Though simple, we found this strategy yielded good results.

## 4 Experimental Setup

We briefly describe the training and evaluation datasets, and the previous XEL approaches from the literature used in our comparison.

### 4.1 Training Mentions

Following previous work, we use hyperlinks from Wikipedia (dumps dated 05/20/2017) as our source of grounded mentions for supervision. Wikipedias in different languages have different pages for the same entity, which are resolved by using inter-language links (e.g., page 利物浦 in Chinese Wikipedia resolves to Liverpool in English). Training mentions statistics are shown in Table 1.

We evaluate on 8 languages – German (de), Spanish (es), Italian (it), French (fr), Chinese (zh), Arabic (ar), Turkish (tr) and Tamil (ta), each of which has varying amount of grounded mentions from the respective Wikipedia (Table 1). We note that our method is applicable to any of the 293 Wikipedia languages as a target language.

### 4.2 Evaluation Datasets

We evaluate XELMS on the following benchmark datasets, spanning 8 different languages, thus providing an extensive evaluation.

| Lang. | # Train Mentions | Size Relative to # English Mentions |
|---|---|---|
| German (de) | 22.6M | 43.7% |
| Spanish (es) | 13.8M | 26.7% |
| French (fr) | 16.2M | 31.3% |
| Italian (it) | 11.5M | 22.2% |
| Chinese (zh) | 5.9M | 11.4% |
| Arabic (ar) | 3.1M | 6.0% |
| Turkish (tr) | 1.8M | 3.5% |
| Tamil (ta) | 473k | 0.9% |

Table 1: Number of train mentions (from Wikipedia) in each language, with % size relative to English (51.7M mentions). Train mentions from Wikipedias like Arabic, Turkish and Tamil are <10% the size of those from the English Wikipedia.

**McN-Test** dataset from (McNamee et al., 2011). The test set was collected by using parallel document collections, and then crowd-sourcing the ground truths. All the test mentions in this dataset consists of person-names only.

**TH-Test** A subset of the dataset used in (Tsai and Roth, 2016), derived from Wikipedia.[3] The mentions in the dataset fall in two categories – *easy* and *hard*, where hard mentions are those for which the most likely candidate according to the prior probability (i.e., $\arg\max \text{P}_{\text{prior}}(e \mid m)$) is *not* the correct title. Indeed, most Wikipedia mentions can be correctly linked by selecting the most likely candidate (Ratinov et al., 2011). We use all the hard mentions from Tsai and Roth (2016)'s test splits for each language, and collectively call this subset TH-TEST.

**TAC15-Test** TAC 2015 (Ji et al., 2015) dataset for Chinese and Spanish. It contains documents from discussion forum articles and news.

We evaluate all models using linking accuracy on gold mentions, and assume gold mentions are provided at test time. Table 2 summarizes the different domains of the evaluation datasets.

**Tuning** We avoid any dataset-specific tuning, instead tuning on a development set and applying the same parameters across all datasets. All tunable parameters were tuned on a development set containing the hard mentions from the train split released by Tsai and Roth (2016). We refer the reader to Appendix B for details on tuning.

[3] Pan et al. (2017) also created a dataset using Wikipedia, but did not categorize mentions like Tsai and Roth (2016). Preliminary experiments on their dataset showed XELMS consistently beat Pan et al. (2017)'s model. We chose TH-TEST for more controlled experiments.

| Dataset | Lang. | Source |
|---|---|---|
| TH-TEST | de, es, fr, it,<br>zh, ar, tr, ta | Wikipedia |
| MCN-TEST | de, es, fr, it,<br>zh, ar, tr | News,<br>Parliament Proceedings |
| TAC15-TEST | es, zh | News,<br>Discussion Forums |

Table 2: Evaluation datasets used in our experiments.

### 4.3 Comparative Approaches

We compare against the following state-of-the-art (SoTA) approaches, described with the language from which they use mention contexts in (.),

**Tsai and Roth (2016) (Target Only)** trains a separate XEL model for each language using mention contexts from the target language Wikipedia only. Current SoTA on TH-TEST.

**Pan et al. (2017) (English Only)** uses entity coherence statistics from English Wikipedia and the document context of a mention for XEL. Current SoTA on MCN-TEST, except for Italian and Turkish, for which it's McNamee et al. (2011).

**Sil et al. (2018) (English Only)** uses multilingual embeddings to transfer a pre-trained English entity linking model to perform XEL for Spanish and Chinese. Prior probabilities $P_{\text{prior}}$ are used as a feature. Current SoTA on TAC15-TEST.

## 5 Experiments

We show that: **(a)** XELMS can train a better entity linking model for a target language on various benchmark datasets by exploiting additional data from a high resource language like English (§5.1). **(b)** XELMS can train a *single* XEL model for multiple related languages and improve upon separately trained models (§5.2). **(c)** Adding additional type information as multi-task loss to XELMS further improves performance (§5.3).

In all tables, we report the linking accuracy of XELMS, averaged over 5 different runs, and mark with * the statistical significance ($p < 0.01$) of the best result (shown **bold**) against the state-of-the-art (SoTA) using Student's one-sample t-test.

### 5.1 Monolingual and Joint Models

In Table 3 and 4 we compare XELMS(mono), which uses monolingual supervision in the target language only, and XELMS(joint), which uses supervision from English in addition to the monolingual supervision, with the state-of-the-art approaches.

| **Dataset** → | TH-TEST | | | MCN-TEST | | |
|---|---|---|---|---|---|---|
| | | XELMS | | | XELMS | |
| **Lang** ↓ | SoTA | mono | joint | SoTA | mono | joint |
| de | 53.3 | 53.7 | **55.6*** | 89.7 | 90.9 | **91.5** |
| es | 54.5 | 54.9 | **56.6*** | **91.5** | 91.2 | 91.4 |
| fr | 47.5 | 48.5 | **49.9*** | 92.1 | 92.6 | **92.7** |
| it | 48.3 | 48.4 | **51.9*** | 85.9 | 87.0 | **87.8*** |
| zh | 57.6 | 58.1 | **61.3*** | **91.2*** | 87.4 | 88.2 |
| ar | 62.1 | 62.6 | **63.8*** | 80.2 | 80.3 | **83.1*** |
| tr | 60.2 | 61.0 | **61.7*** | **95.3*** | 91.0 | 91.9 |
| ta | 54.1 | 54.7 | **59.7*** | n/a | n/a | n/a |
| avg. | 54.7 | 55.2 | **57.6** | 89.4 | 88.6 | **89.5** |

Table 3: XELMS(joint) improves upon XELMS(mono) and the current State-of-The-Art (SoTA) on TH-TEST and MCN-TEST, showing the benefit of using additional supervision from English. The best score is shown **bold** and * marks statistical significance of best against SoTA. Refer §4.3 for details on SoTA.

| | **Model** ↓ **Lang.** → | es | zh |
|---|---|---|---|
| | (Tsai and Roth, 2016) | 82.4 | 85.1 |
| | (Sil et al., 2018) (SoTA) | 83.9 | 85.9 |
| XELMS | mono | 83.3 | 84.4 |
| | mono$^{+\text{type}}$ | 83.5 | 84.8 |
| | joint | 84.1 | 85.5 |
| | joint$^{+\text{type}}$ | **84.4*** | **86.0** |
| | multi | 83.9 | n/a |
| | multi$^{+\text{type}}$ | **84.4*** | n/a |

Table 4: Linking accuracy on TAC15-Test. Numbers for Sil et al. (2018) from personal communication.

We see that XELMS(mono) achieves similar or slightly better scores than respective SoTA on all datasets. The SoTA for MCN-TEST in Turkish and Chinese enhances the model by using transliteration for candidate generation, explaining their superior performance. XELMS(joint) performs substantially better than XELMS(mono) on all datasets (Table 3 and 4), proving that using additional supervision from a high resource language like English leads to better linking performance. In particular, XELMS(joint) outperforms the SoTA on all languages in TH-TEST, on Spanish in TAC15-Test, and on 4 of the 7 languages in MCN-TEST.

### 5.2 Multilingual Training

XELMS is the first approach that can train a *single* XEL model for multiple languages. To demonstrate this capability, we train a model, henceforth referred as XELMS(multi), *jointly* on 5 related languages – Spanish, German, French, Italian and En-

| Dataset → | TH-TEST | | | MCN-TEST | | |
|---|---|---|---|---|---|---|
| | | XELMS | | | XELMS | |
| Lang ↓ | SoTA | joint | multi | SoTA | joint | multi |
| de | 53.3 | **55.6**$^*$ | 55.2 | 89.7 | **91.5** | 91.4 |
| es | 54.5 | 56.6 | **56.8**$^*$ | **91.5** | 91.4 | 91.4 |
| fr | 47.5 | 49.9 | **51.0**$^*$ | 92.1 | **92.7** | 92.6 |
| it | 48.3 | 51.9 | **52.3**$^*$ | 85.9 | 87.8 | **87.9**$^*$ |
| avg. | 50.9 | 53.5 | **53.8** | 89.8 | **90.8** | **90.8** |

Table 5: Linking accuracy of a *single* XELMS(multi) model for four languages – German, Spanish, French and Italian. Individually trained XELMS(joint) scores are also shown. The best score is shown **bold** and $^*$ marks statistical significance of **best** against SoTA. Refer §4.3 for details on SoTA.

glish. We compare XELMS(multi) to the respective XELMS(joint) model for each language.

Table 4 and 5, show that XELMS(multi) is better (or at par) than XELMS(joint) on all datasets. This shows that XELMS(multi) can making more efficient use of available supervision in related languages than previous approaches which trained separate models per language.

### 5.3 Adding Fine-grained Type Information

To study the effect of adding fine-grained type information, in Table 4 we compare XELMS(mono) and XELMS(joint) to XELMS(mono$^{+type}$) and XELMS(joint$^{+type}$) respectively, which are versions of XELMS(mono) and XELMS(joint) trained using the two type-aware losses.

XELMS(mono$^{+type}$) and XELMS(joint$^{+type}$) both improve compared to XELMS(mono) and XELMS(joint) on MCN-TEST and TH-TEST (Table 6 vs Table 3), showing the benefit of using structured knowledge in the form of fine-grained types. Similar trends are also seen on TAC15-TEST (Table 4), where XELMS(joint$^{+type}$) improves on the SoTA for Spanish and Chinese.

## 6 Experiments with Limited Resources

The key motivation of XELMS is to exploit supervision from high-resource languages like English to aid XEL for languages with limited resources. In this section, we examine two such scenarios,

**(a)** *Zero-shot setting* i.e., no supervision available in the target language. Our analysis reveals the limitations of zero-shot XEL approaches and finds that the prior probabilities play an important role in achieving good performance (§6.1), which are unavailable in realistic zero-shot scenarios.

**(b)** *Low-resource setting* i.e., some supervision available in the target language. We show that by combining supervision from a high-resource language, like English, XELMS can achieve competitive performance with a fraction of available supervision in the target language (§6.2).

| Dataset → | TH-TEST | | | MCN-TEST | | |
|---|---|---|---|---|---|---|
| | | XELMS | | | XELMS | |
| Lang ↓ | SoTA | mono$^{+type}$ | joint$^{+type}$ | SoTA | mono$^{+type}$ | joint$^{+type}$ |
| de | 53.3 | 54.0 | **55.9**$^*$ | 89.7 | 91.2 | **91.5** |
| es | 54.5 | 55.1 | **57.2**$^*$ | **91.5** | 91.0 | 91.2 |
| fr | 47.5 | 49.0 | **50.6**$^*$ | 92.1 | 92.6 | **92.7** |
| it | 48.3 | 49.2 | **52.2**$^*$ | 85.9 | 87.4 | **87.9**$^*$ |
| zh | 57.6 | 58.9 | **61.5**$^*$ | **91.2**$^*$ | 87.6 | 88.4 |
| ar | 62.1 | 63.0 | **64.0**$^*$ | 80.2 | 81.1 | **84.0**$^*$ |
| tr | 60.2 | 61.5 | **62.0**$^*$ | **95.3**$^*$ | 91.2 | 92.1 |
| ta | 54.1 | 56.0 | **59.9**$^*$ | n/a | n/a | n/a |
| avg. | 54.7 | 55.8 | **57.9** | 89.4 | 88.9 | **89.7** |

Table 6: Adding fine-grained type information further improves linking accuracy (compare to Table 3). The best score is shown **bold** and $^*$ marks statistical significance of best against SoTA. Refer §4.3 for details on SoTA.

### 6.1 Zero-shot Setting

We first explain how XELMS can perform zero-shot XEL, the implications of our zero-shot setting, and how it is more realistic than previous work.

**Zero-shot XEL with XELMS** XELMS performs zero-shot XEL by training a model using English supervision and multilingual embeddings for English, and directly applying it to the test data in another language using the respective multilingual word embedding instead of English embeddings.

**No Prior Probabilities** Prior probabilities (or prior), i.e., $P_{prior}$ have been shown to be a reliable indicator of the correct disambiguation in entity linking (Ratinov et al., 2011; Tsai and Roth, 2016). These probabilities are estimated from counts over the training mentions in the target language. In the absence of training data for the target language, as in the zero-shot setting, these prior probabilities are not available to an XEL model.

**Comparison to Previous Work** The only other model capable of zero-shot XEL is that of Sil et al. (2018). However, Sil et al. (2018) use prior probabilities and coreference chains for the target language in their zero-shot experiments, both of which will not be available in a realistic zero-shot scenario. Compared to Sil et al. (2018), we evaluate the performance of zero-shot XEL in more realistic setting, and show it is adversely affected by absence of prior probabilities.

| Dataset →<br>Approach ↓ | TAC15-Test<br>(es) | <br>(zh) | TH-Test<br>(avg) | McN-Test<br>(avg) |
|---|---|---|---|---|
| XELMS (Z-S w/ prior) | 80.3 | 83.9 | 43.5 | 88.1 |
| XELMS (Z-S w/o prior) | 53.5 | 55.9 | 41.1 | 86.0 |
| SoTA | 83.9 | 85.9 | 54.7 | 89.4 |

Table 7: Linking accuracy of the zero-shot (Z-S) approach on different datasets. Zero-shot (w/ prior) is close to SoTA for datasets like TAC15-Test, but performance drops in the more realistic setting of zero-shot (w/o prior) (§6.1) on all datasets, indicating most of the performance can be attributed to the presence of prior probabilities. The slight drop in MCN-TEST is due to trivial mentions, which only have a single candidate.

**Is zero-shot XEL really effective?** To evaluate the effectiveness of the zero-shot XEL approach, we perform zero-shot XEL using XELMS on all datasets. Table 7 shows zero-shot XEL results on all datasets, both with and without using the prior during inference. Note that zero-shot XEL (with prior) is close to SoTA (Sil et al. (2018)) on TAC15-TEST, which also uses the prior for zero-shot XEL. However, for zero-shot XEL (without prior) performance drops by more than 20% for TAC15-Test, 2.4% for TH-Test and by 2.1% for McN-Test. This indicates that zero-shot XEL is not effective in a realistic zero-shot setting (i.e., when the prior is unavailable for inference).

We found that the prior is indeed a strong indicator of the correct disambiguation. For instance, simply selecting the the most likely candidate using the prior for TAC15-TEST achieved 77.2% and 78.8% for Spanish and Chinese respectively. It is interesting to note that both zero-shot XEL (with or without prior) perform worse than the best possible model on TH-TEST, because TH-TEST was constructed to ensure prior probabilities are not strong indicators (Tsai and Roth, 2016). On MCN-TEST, we found that an average of 75.9% mentions have only one (the correct) candidate, making them trivial to link, regardless of the absence of priors.

The results show that most of the XEL performance in zero-shot settings can be attributed to availability of prior probabilities for the candidates. It is evident that zero-shot XEL in a realistic setting (i.e., when prior probabilities are not available) is still a challenging problem.

## 6.2 Low-resource Setting

We analyze the behavior of XELMS in a low-resource setting, i.e. when some supervision is available in the target language. The aim of this setting is to estimate how much supervision from the target language is needed to get reasonable performance when using it jointly with supervision from English. To discount the effect of prior probabilities, we report all results without the prior.

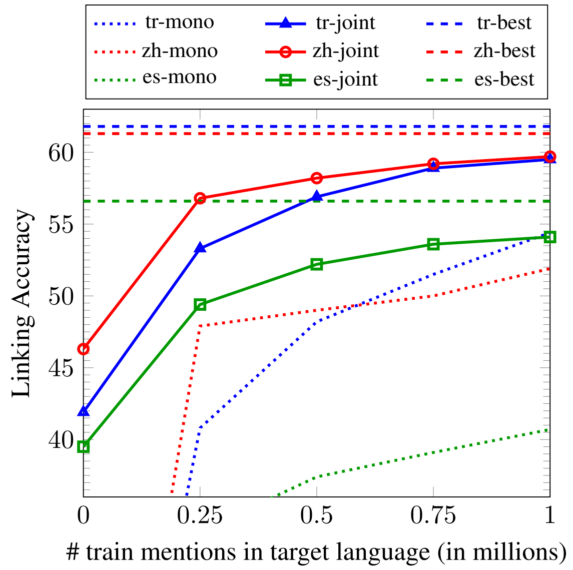


Figure 4: Linking accuracy vs. the number of train mentions in the target language L (= Turkish (tr), Chinese (zh) and Spanish (es)). We compare both XELMS(mono) and XELMS(joint) to the best results using all available supervision, denoted by L-best. To discount the effect of the prior, all results above are without it. For number of train mentions = 0, XELMS(joint) is equivalent to zero-shot without prior. Best viewed in color.

Figure 4 plots results on the TH-Test dataset when training a XELMS(joint) model by gradually increasing the number of mention contexts for target language L (= Spanish, Chinese and Turkish) that are available for supervision. Figure 4 also shows the best results achieved using all available target language supervision (denoted by L-best). For comparison with the mono-lingually supervised model, we also plot the performance of XELMS(mono), which only uses the target language supervision.

Figure 4 shows that after training on 0.75M mentions from Turkish and Chinese (and 1.0M mentions from Spanish), the XELMS(joint) model is within 2-3% of the respective L-best model which uses all training mentions in the target language, indicating that XELMS(joint) can reach competitive performance even with a fraction of the full target language supervision. For comparison, a XELMS(mono) model trained on the same number of training mentions is 5-10% behind the respective XELMS(joint) model, showing better utilization of target language supervision by XELMS(joint).

## 7 Related Work

Existing approaches have taken two main directions to obtain supervision for learning XEL models — **(a)** using mention contexts appearing in the target language (McNamee et al., 2011; Tsai and Roth, 2016), or **(b)** using mention contexts appearing only in English (Pan et al., 2017; Sil et al., 2018). We describe these directions and their limitations below, and explain how XELMS overcomes these limitations.

McNamee et al. (2011) use annotation projection via parallel corpora to generate mention contexts in the target language, while Tsai and Roth (2016) learns separate XEL models for each language and only use mention contexts in the target language. Both these approach have scalability issues for languages with limited resources. Another limitation of these approaches is that they train separate models for each language, which is inefficient when working with multiple languages. XELMS overcomes these limitations as it can use mention context from multiple languages simultaneously, and train a single model.

Other approaches only use mention contexts from English. While Pan et al. (2017) compute entity coherence statistics from English Wikipedia, Sil et al. (2018) perform zero-shot XEL for Chinese and Spanish by using multilingual embeddings to transfer a pre-trained English EL model. However, our work suggests that mention contexts in the target language should also be used, if available. Indeed, a recent study (Lewoniewski et al., 2017) found that for language sensitive topics, the quality of information can be better in the relevant language version of Wikipedia than the English version. Our analysis also shows that zero-shot XEL approaches like that of Sil et al. (2018) are not effective in realistic zero-shot scenarios where good prior probabilities are unlikely to be available. In such cases, we showed that combining supervision available in the target language with supervision from a high-resource language like English can yield significant performance improvements.

The architecture of XELMS is inspired by several monolingual entity linking systems (Francis-Landau et al., 2016; Nguyen et al., 2016; Gupta et al., 2017), approaches that use type information to aid entity linking (Ling et al., 2015; Gupta et al., 2017; Raiman and Raiman, 2018), and the recent success of multilingual embeddings for several tasks (Ammar et al., 2016a; Duong et al., 2017).

## 8 Conclusion

We introduced XELMS, an approach that can combine supervision from multiple languages to train an XEL model. We illustrate its benefits through extensive evaluation on different benchmarks. XELMS is also the first approach that can train a *single* model for multiple languages, making more efficient use of available supervision than previous approaches which trained separate models.

Our analysis sheds light on the poor performance of zero-shot XEL in realistic scenarios where the prior probabilities for candidates are unlikely to exist, in contrast to findings in previous work that focused on high-resource languages. We also show how in low-resource settings, XELMS makes it possible to achieve competitive performance even when only a fraction of the available supervision in the target language is provided.

Several future research directions remain open. For all XEL approaches, the task of candidate generation is currently limited by existence of a target language Wikipedia and remains a key challenge. A joint inference framework which enforces coherent predictions (Cheng and Roth, 2013; Globerson et al., 2016; Ganea and Hofmann, 2017) could also lead to further improvements for XEL. Similar techniques can be applied to other information extraction tasks like relation extraction to extend them to multilingual settings.

## Acknowledgments

The authors would like to thank Snigdha Chaturvedi, Anne Cocos, Stephen Mayhew, Chen-Tse Tsai, Qiang Ning, Jordan Kodner, Dan Deutsch, John Hewitt and the anonymous EMNLP reviewers for their useful comments and suggestions.

This work was supported by Contract HR0011-15-2-0025 and Agreement HR0011-15-2-0023 with the US Defense Advanced Research Projects Agency (DARPA). Approved for Public Release, Distribution Unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

## References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016a. Many Languages, One Parser. In *Transactions of the Association for Computational Linguistics*, volume 4.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. 2016b. Massively Multilingual Word Embeddings. *arXiv preprint arXiv:1602.01925*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. In *Transactions of the Association for Computational Linguistics*, volume 5.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proc. of ACM SIGMOD*.

Rich Caruana. 1998. Multitask Learning. In *Learning to Learn*, pages 95–133. Springer.

Xiao Cheng and Dan Roth. 2013. Relational Inference for Wikification. In *Proc. of EMNLP*.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2017. Multilingual Training of Crosslingual Word Embeddings. In *Proc. of EACL*.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks. In *Proc. of NAACL-HLT*.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep Joint Entity Disambiguation with Local Neural Attention. In *Proc. of EMNLP*.

Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringaard, and Fernando Pereira. 2016. Collective Entity Resolution with Multi-Focal Attention. In *Proc. of ACL*.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity Linking via Joint Encoding of Types, Descriptions, and Context. In *Proc. of EMNLP*.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. In *Proc. of IJCAI*.

Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking. In *Text Analysis Conference*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *Proc. of CVPR*.

Włodzimierz Lewoniewski, Krzysztof Wecel, and Witold Abramowicz. 2017. Relative Quality and Popularity Evaluation of Multilingual Wikipedia Articles. In *Journal of Informatics*.

Xiao Ling, Sameer Singh, and Daniel S Weld. 2015. Design Challenges for Entity Linking. In *Transactions of the Association for Computational Linguistics*, volume 3.

Xiao Ling and Daniel S Weld. 2012. Fine-grained Entity Recognition. In *Proc. of AAAI*.

Paul McNamee, James Mayfield, Dawn Lawrie, Douglas W Oard, and David S Doermann. 2011. Cross-Language Entity Linking. In *Proc. of IJCNLP*.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: Linking Documents to Encyclopedic Knowledge. In *Proc. of CIKM*.

Thien Huu Nguyen, Nicolas Fauceglia, Mariano Rodriguez Muro, Oktie Hassanzadeh, Alfio Massimiliano Gliozzo, and Mohammad Sadoghi. 2016. Joint Learning of Local and Global Features for Entity Linking via Neural Networks. In *Proc. of COLING*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual Name Tagging and Linking for 282 Languages. In *Proc. of ACL*.

Jonathan Raiman and Olivier Raiman. 2018. DeepType: Multilingual Entity Linking by Neural Type System Evolution. In *Proc. of AAAI*.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proc. of ACL-HLT*.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural Cross-lingual Entity Linking. In *Proc. of AAAI*.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline Bilingual Word Vectors, Orthogonal Transformations, and the Inverted Softmax. In *Proc. of ICLR*.

Valentin I. Spitkovsky and Angel X. Chang. 2012. A Cross-Lingual Dictionary for English Wikipedia Concepts. In *Proc. of LREC*.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual Wikification Using Multilingual Embeddings. In *Proc. of NAACL*.

Chen-Tse Tsai and Dan Roth. 2018. Learning Better Name Translation for Cross-Lingual Wikification. In *Proc. of AAAI*.

Shyam Upadhyay, Jordan Kodner, and Dan Roth. 2018. Bootstrapping Transliteration with Constrained Discovery for Low-Resource Languages. In *Proc. of EMNLP*.