Towards Zero-resource Cross-lingual Entity Linking

Shuyan Zhou, Shruti Rijhwani, Graham Neubig

Language Technologies Institute
Carnegie Mellon University
{shuyanzh,srijhwan,gneubig}@cs.cmu.edu

Abstract

Cross-lingual entity linking (XEL) grounds named entities in a source language to an English Knowledge Base (KB), such as Wikipedia. XEL is challenging for most languages because of limited availability of requisite resources. However, much previous work on XEL has been on simulated settings that actually use significant resources (e.g. source language Wikipedia, bilingual entity maps, multilingual embeddings) that are unavailable in truly low-resource languages. In this work, we first examine the effect of these resource assumptions and quantify how much the availability of these resource affects overall quality of existing XEL systems. Next, we propose three improvements to both entity candidate generation and disambiguation that make better use of the limited data we do have in resource-scarce scenarios. With experiments on four extremely low-resource languages, we show that our model results in gains of 6-23% in end-to-end linking accuracy.[1]

1 Introduction

Entity linking (EL; Bunescu and Paşca (2006); Cucerzan (2007); Dredze et al. (2010); Hoffart et al. (2011)) identifies entity mentions in a document and associates them with their corresponding entries in a structured Knowledge Base (KB) (Shen et al., 2015), such as Wikipedia or Freebase (Bollacker et al., 2008). EL involves two main steps: (1) *candidate generation*, retrieving a list of candidate KB entries for each entity mention, and (2) *disambiguation*, selecting the most likely entry from the candidate list.

In this work, we focus on cross-lingual entity linking (XEL; McNamee et al. (2011), Ji et al. (2015)), where the document is in a (source) language that is different from the (target) language

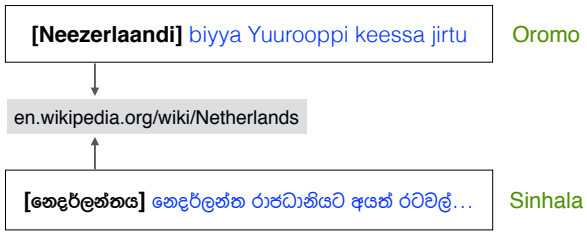

Figure 1: XEL for two low-resource languages – Oromo and Sinhala, linking source mentions to entity "Netherlands" in English Wikipedia.

of the KB. Following recent work (Sil et al., 2018; Upadhyay et al., 2018), we use English Wikipedia as this KB. Figure 1 shows an example.

XEL to English from major languages such Spanish and Chinese has been carefully studied, and significant progress has been made. Success in these languages can be largely attributed to the availability of rich resources. Specifically, the following is a list of resources required by recent works (Tsai and Roth, 2016; Pan et al., 2017; Sil et al., 2018; Upadhyay et al., 2018):

English Wikipedia ($\mathbb{W}_{\text{eng}}$): The target KB and a large corpus of text. Importantly, the text is annotated with anchor text linking between entity mentions (e.g. "Holland" in the body text of an article) and the page for the entity (e.g. "Netherlands"). These annotations can be used to extract mention-entity maps for entity candidate generation, and to directly train entity disambiguation systems.

Source Language Wikipedia ($\mathbb{W}_{\text{src}}$): KB and corresponding text in the source language. Similarly to English Wikipedia, this can be used to obtain mention-entity maps or train disambiguation systems, but the size of Wikipedia is relatively small for most low-resource languages.

Bilingual Entity Maps ($\mathbb{M}$): A map between source language entities and English entities. One common source of this map is Wikipedia inter-language links between the source language and English. These inter-language links can directly

[1]Code is available at https://github.com/shuyanzhou/burn_xel

Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo), pages 243–252
Hong Kong, China, November 3, 2019.
https://doi.org/10.18653/v1/P17

and unambiguously link entities in the source language KB to the English KB.

Multilingual Embeddings ($\mathbb{E}$): These embeddings map words in different languages to the same vector space.

The availability of these resources varies widely among languages. They are available for high-resource languages such as Spanish and Chinese, which have been widely used as test-beds for XEL. For example, there are over 1.5 million articles in Spanish Wikipedia, which provide an abundance of annotations. However, the situation is not as favorable for most other languages: while $\mathbb{W}_{\text{eng}}$ is invariant of the source language to link from, many of the other resources are small or non-existent. In fact, only 300 languages (from ≈7000 living languages in the world) have Wikipedia $\mathbb{W}_{\text{src}}$, and among these many have a limited number of pages. For example, Oromo, a Cushitic language with 30 million speakers, has only 776 Wikipedia pages. It is similarly difficult to obtain exhaustive bilingual entity maps, and for many languages even the monolingual/parallel text necessary to train multilingual embeddings is scarce.

This work makes two major contributions regarding XEL for low-resource languages.

The first major contribution is empirical. We extensively evaluate the effect of resource restrictions on existing XEL methods in true low-resource settings instead of simulated ones (Section 4). We compare the performance of both the candidate generation model and the disambiguation model of our baseline XEL system between two high-resource languages and four low-resource languages. We quantify how much the availability of the aforementioned resources affect the overall quality of the existing methods, and find that *with scarce access to these resources, the performance of existing methods drops significantly*. This highlights the effect of resource constraints in realistic settings, and indicates that these constraints should be considered more carefully in future system design.

Our second major contribution is methodological. We propose three methods as first steps towards ameliorating the large degradation in performance we see in low-resource settings. (1) We investigate a *hybrid candidate generation method*, combining existing lookup-based and neural candidate generation methods to improve candidate list recall by 9-24%. (2) We propose a set of *entity disambiguation features that are entirely language-agnostic*, allowing us to train a disambiguation system on English and transfer it directly to low-resource languages. (3) We design a *non-linear feature combination* method, which makes it possible to combine features in a more flexible way. We test these three methodological improvements on four extremely low-resource languages (Oromo, Tigrinya, Kinyarwanda, and Sinhala), and find that the combination of these three techniques leads to consistent performance gains in all four languages, amounting to 6-23% improvement in end-to-end XEL accuracy.

2 Problem Formulation

Given a set of documents $\mathcal{D} = \{D_1, D_2, ..., D_l\}$ in any source language L_s, a set of detected mentions $\mathbf{M}_D = \{m_1, m_2, ..., m_n\}$ for each document D, and the English Wikipedia $\mathbf{E}_{\text{KB}}$, the goal of XEL is to associate each mention with its corresponding entity in the English Wikipedia. We denote an entity in English Wikipedia as e and its parallel entity in the source language Wikipedia as e^{src}.

For each $m_i \in \mathbf{M}_D$, candidate generation first retrieves a list of candidate entities $\mathbf{e}_i = \{e_{i,1}, e_{i,2}, ..., e_{i,n}\}$ from $\mathbf{E}_{\text{KB}}$ based on probabilities $\mathbf{p}_i = \{p_{i,1}, p_{i,2}, ..., p_{i,n}\}$ where $p_{i,j}$ denotes $p(e_{i,j}|m_i)$. Then, the disambiguation model assigns a score $s(e_{i,j}|D)$ to each $e_{i,j}$. These scores are normalized among $\mathbf{e}_i$ and result in the probability $p(e_{i,j}|D)$. The entity with highest score is selected as the prediction. We denote the gold entity as e^*.

Performance of candidate generation is measured by *gold candidate recall*: the proportion of mentions whose top-n candidate list contains the gold entity over all test mentions. This recall upper-bounds performance of an entity disambiguation system. In the consideration of the computational cost of the more complicated downstream disambiguation model, this n is often 30 or smaller (Sil et al., 2018; Upadhyay et al., 2018). The performance of an end-to-end XEL system is measured by *accuracy*: the proportion of mentions whose predictions are correct. We follow Yamada et al. (2017); Ganea and Hofmann (2017) and focus on *in-KB* accuracy; we ignore mentions whose linked entity does not exist in the KB in this work.

3 Baseline Model

This section describes existing methods for candidate generation and disambiguation, and our baseline XEL system, which is heavily inspired by existing works (Ling et al., 2015; Globerson et al., 2016; Pan et al., 2017). We investigate the effect of resource constraints on this system in Section 4. Based on empirical observations, we propose our improved XEL system in Section 5 and present its results in Section 6.

3.1 Candidate Generation

WIKIMENTION: With access to all the resources we list above, there is a straightforward approach to candidate generation used by most state-of-the-art work in XEL (Sil et al., 2018; Upadhyay et al., 2018). Specifically, a monolingual mention-entity map can be extracted from $\mathbb{W}_{\text{src}}$ by finding all cross-article links in $\mathbb{W}_{\text{src}}$, and using the anchor text as mention m and the linked entity as e^{src}. These entities are then redirected to English Wikipedia with $\mathbb{M}$ to obtain e. For instance, if Oromo mention "Itoophiyaatti" is linked to entity "Itoophiyaa" in some Oromo Wikipedia pages, the corresponding English Wikipedia entity "Ethiopia" will be acquired through $\mathbb{M}$ and used as a candidate entity for the mention. The score $p(e_{i,j}|m_i)$ provided by this model shows the *probability* of linking to $e_{i,j}$ when mentioning m_i. Because of its heavy reliance on $\mathbb{W}_{\text{src}}$ and $\mathbb{M}$, WIKIMENTION does not generalize well to real low-resource settings. We discuss this in Section 4.1.

PIVOTING: Recently, Rijhwani et al. (2019) propose a zero-shot transfer learning method for XEL candidate generation, which uses no resources in the source language. A character-level LSTM is trained to encode entities using a bilingual entity map between some high-resource language and English. If the chosen high-resource language is closely related to the low-resource language (same language family, shared orthography etc.), zero-shot transfer will often be successful in generating candidates for the low-resource language. In this case, the model generated score $s(e_{i,j}|m_i)$ indicates the *similarity* which should be further normalized into a *probability* $p(e_{i,j}|m_i)$ (Section 5.1).

Notably, both methods have advantages and disadvantages, with PIVOTING generally being more robust, and WIKIMENTION being more accurate when resources are available. To take advantage of this, we propose a method for calibrated combination of these two methods in Section 5.1.

3.2 Featurization and Linear Scoring

Next, we move to the entity disambiguation step, which we further decompose into (1) the design of features and (2) the choice of inference model that combines these features together.

3.2.1 Featurization

Unfortunately for low-resource settings, many XEL disambiguation models rely on extensive resources such as $\mathbb{E}$ and $\mathbb{W}_{\text{src}}$ (Sil et al., 2018; Upadhyay et al., 2018) to obtain features. However, some previous work on XEL does limit its resource usage to $\mathbb{W}_{\text{eng}}$, which is available regardless of the source language. Our baseline follows one such method by Pan et al. (2017).

We use two varieties of features: *unary* features that reflect properties of a single entity and *binary* features that quantify coherence between pairs of entities. The top half of Table 1 shows unary feature functions, which take one argument $e_{i,j}$ and return a value that represents some property of this entity. The grayed mention-entity prior $f_l^1(e_{i,j})$ is the main unary feature used by Pan et al. (2017), and we use this in our baseline. Binary features are in the bottom half of Table 1. Each binary feature function $f_g^i(e_{i,j}, e_{k,w})$ takes two entities as arguments, and returns a value that indicates the relatedness between the entities. Similarly, the grayed co-occurrence feature $f_g^1(e_{i,j}, e_{k,w})$ is used in the baseline. We refer to these two features as BASE.

While these features have proven useful in higher-resource XEL, in lower-resource scenarios, we hypothesize that it is more important to design features that make the most use of the language-invariant resource $\mathbb{W}_{\text{eng}}$ to make up for the relative lack of other resources in the source language. We discuss more intelligent features in Section 5.2.

3.2.2 Non-iterative Linear Inference Model

While the design of features is resource-sensitive, the choice of an inference model is fortunately resource-agnostic as it only relies on the existence of features. Our baseline follows existing (X)EL works (Ling et al., 2015; Globerson et al., 2016; Pan et al., 2017) to *linearly* aggregate unary features to a *local* score $s_l(e|D)$ and binary features to a *global* score $s_g(e|D)$. The local score reflects the properties of an independent entity, and the global score quantifies the coherence between an

entity and other linked entities in the document. The score of each entity is defined as:

$$s(e_{i,j}|D) = s_g(e_{i,j}|D) + s_l(e_{i,j}|D)$$

The local score is the linear combination of unary features $f_l^i(e_{i,j}) \in \mathbf{\Phi}(e_{i,j})$:

$$s_l(e_{i,j}|D) = \mathbf{W}_l^T \mathbf{\Phi}(e_{i,j}) \quad (1)$$

where $\mathbf{W}_l \in \mathbb{R}^{d_l \times 1}$ and d_l is the number of unary features in the vector.

On the other hand, the global score s_g is an average aggregation of mention evidence s_m across the document. Each $s_m(m_k, e_{i,j})$ indicates how strongly a context mention m_k supports the j-th candidate entity of mention m_i:

$$s_g(e_{i,j}|D) = \frac{1}{|\mathbf{M}_D|} \sum_{k \neq i} s_m(m_k, e_{i,j}) \quad (2)$$

As a mention is in fact the surface form of other candidate entities, $s_m(m_k, e_{i,j})$ can be measured by the relatedness between the candidate entities $\mathbf{e}_k$ of m_k and $e_{i,j}$. Our baseline inference model follows Ling et al. (2015); Globerson et al. (2016) to process this evidence in a GREEDY manner:

$$s_m(m_k, e_{i,j}) = \max_{e_{k,w} \in \mathbf{E}_k} (s_e(e_{i,j}, e_{k,w})) \quad (3)$$

Similarly to s_l, $s_e(e_{i,j}, e_{k,w})$ is the linear combination of binary features $f_g^i(e_{i,j}, e_{k,w}) \in \mathbf{\Psi}(e_{i,j}, e_{k,w})$:

$$s_e(e_{i,j}, e_{k,w}) = \mathbf{W}_g^T \mathbf{\Psi}(e_{i,j}, e_{k,w}) \quad (4)$$

The greedy strategy often results in a sub-optimal assignment, as the confidence of each candidate entity is not taken into consideration. To solve this problem, we propose iteratively updating belief of each candidate entity in Section 5.3.

Following Upadhyay et al. (2018); Sil et al. (2018), we consider WIKIMENTION as the baseline candidate generation model and BASE+GREEDY as the baseline disambiguator. We denote WIKIMENTION+BASE+GREEDY as the end-to-end baseline system.

4 Experiment I: Real Low-resource Constraints in XEL

In this section, we study the effects of resource constraints in truly low-resource settings; we then evaluate how this changes the conclusions we may draw about the efficacy of existing XEL models. We attempt to answer the following research questions: (1) how the does the availability of resources influence the performance of XEL systems, and (2) how do truly low-resource settings diverge from XEL with more resources?

We perform this study within the context of our WIKIMENTION+BASE+GREEDY baseline (which is conceptually similar to previous work). We carry out the study on several languages and datasets:

TAC-KBP: TAC-KBP 2011 for English (*en*) (Ji et al., 2011), TAC-KBP 2015 for Spanish (*es*) and Chinese (*zh*) (Ji et al., 2015). All contain documents from forums and news.

DARPA-LRL: The DARPA LORELEI annotated documents[2] in 4 low-resource languages: Tigrinya (*ti*), Oromo (*om*), Kinyarwanda (*rw*) and Sinhala (*si*). These are news articles, blogs and social media posts about disasters and humanitarian crises.

Detailed experimental settings are in Section 6.1. It is notable that a large number of previous works examine XEL on simulated low-resource settings such as the TAC-KBP datasets for large languages such as Chinese and English (Sil et al., 2018; Upadhyay et al., 2018), while the DARPA-LRL datasets are more reflective of true constraints in low-resource scenarios.

4.1 Results

Table 2 shows various statistics for the baseline system on English, two high-resource, and four low-resource XEL languages. The first row of Table 2 shows the gold candidate recall of WIKIMENTION on 7 languages. The Wikipedia sizes of each language are shown in the last row of the table for reference. In general, the gold candidate recall of WIKIMENTION is positively correlated with the size of available Wikipedia resoruces. We can note that compared to the four low-resource languages, the statistics of the two high-resource languages are closer to those of English.

End-to-end performance of a system that selects the entity with the highest score according to WIKIMENTION is listed in the second row of the table. This trivial context-insensitive disambiguation method results in performance not far from the upper bound in six XEL languages. However, the size of the gap between this method and

[2] https://www.darpa.mil/program/low-resource-languages-for-emergent-incidents

Symbol	Feature Name	Equation	Resource
$f_l^1(e_{i,j})$	Mention-entity prior score	$\log(\max(p(e_{i,j}\|m_i), \epsilon))$	Variable
$f_l^2(e_{i,j})$	Entity prior	$\log(\max(\frac{c(e_{i,j})}{\sum_{e\in\mathbf{E}_{\text{KB}}} c(e)}, \epsilon))$	$\mathbb{W}_{\text{eng}}$
$f_l^3(e_{i,j})$	Related mention number	$\sum_{m_k\in\mathbf{M}_D\setminus m_i} \mathbb{1}(\text{any}_{e_{k,m}\in\mathbf{E}_k} f_g^1(e_{i,j}, e_{k,m}) > 0)$	-
$f_l^4(e_{i,j})$	Exact match number	$\sum_{m_k\in\mathbf{M}_D\setminus m_i} \mathbb{1}(e \in \mathbf{E}_k)$	-
$f_g^1(e_{i,j}, e_{k,w})$	Co-occurrence probability	$\log(\max(\frac{c(e_{i,j}, e_{k,w})}{c(e_{i,j})}), \epsilon)$	$\mathbb{W}_{\text{eng}}$
$f_g^2 e_{i,j}, e_{k,w})$	Positive Pointwise Mutual Information (PPMI)	$\max(\log_2(\frac{p(e_{i,j}, e_{k,w})}{p'(e_{i,j})p'(e_{k,w})}), 0)$	$\mathbb{W}_{\text{eng}}$
$f_g^3(e_{i,j}, e_{k,w})$	Entity embedding similarity	$\text{cosine}(\mathbf{V}_{e_{i,j}}, \mathbf{V}_{e_{k,w}})$	$\mathbb{W}_{\text{eng}}$
$f_g^4(e_{i,j}, e_{k,w})$	Hyperlink count	$\log(\max(\frac{\sum_{e_k\in\mathbf{H}_{e_{i,j}}} \mathbb{1}(e_{i,j}=e_{k,w})}{\lvert\mathbf{H}_{e_{i,j}}\rvert}, \epsilon)$	$\mathbb{W}_{\text{eng}}$

Table 1: Unary features (top half) and binary features (bottom half). Gray indicates BASE features. "Variable" means this feature comes from the candidate generation model and thus its resource dependency will be decided by that model; ϵ is set to `1e-7`; $c(e)$ is the frequency of an entity among all anchor links in $\mathbb{W}_{\text{eng}}$; $c(e_i, e_j)$ is the co-occurrence count of two entities in $\mathbb{W}_{\text{eng}}$; $p(e_i, e_j)$ is normalized over all entity pairs and $p'(e_i)$ is normalized over all entities with smoothing parameter $\gamma = 0.75$; $\mathbf{V}_e$ represents the entity embedding of e_i; $\mathbf{H}_{e_i}$ represents a set of entities in e_i's English Wikipedia page.

		high-resource		*low-resource*			
Model	en	zh	es	ti	om	rw	si
Gold Candidate Recall	92.4	89.2	89.0	21.9	45.3	45.6	66.6
$p(e\|m)$	70.1	83.1	78.2	21.5	41.0	45.1	63.1
BASE+GREEDY	77.5	85.5	82.9	21.8	38.4	44.9	64.4
Wikipedia Size	5.0M	1.0M	1.5M	168	775	1.8K	15.1K

Table 2: Gold candidate recall of WIKIMENTION over seven languages, accuracy (%) of selecting the highest score entity, and accuracy after end-to-end EL using the BASE+GREEDY method.

the upper bound is largely different between high- and low-resource settings – this gap is significant for high-resource languages, but quite small for the four low-resource languages. Accordingly, in third row where we apply the disambiguation method BASE+GREEDY, we find gains of 2-7% on the high-resource languages, but little to no gain on the low-resource languages. This shows that when using a standard candidate generation method such as WIKIMENTION, there is *little room for more sophisticated disambiguation models to improve performance*, despite the fact that development of disambiguation methods (rather than candidate generation) has been the focus of much prior work.

5 Proposed Model Improvements

Next, we introduce our proposed methods: (1) calibrated combination of two existing candidate generation models, (2) an XEL disambiguation model that makes best use of resources that *will* be available in extremely low-resource settings.

5.1 Calibrated Candidate List Combination

As the gold candidate recall decides the upper bound of an (X)EL system, candidate lists with close to 100% recall are ideal. However, this is hard to achieve for most low-resource languages where existing candidate generation models only provide candidate lists with low recall (less than 60%, as we show in Section 4.1). Further, combination of candidate lists retrieved by different models is non-trivial as the scores are not comparable among models. For example, scores of WIKIMENTION have probabilistic interpretation while scores of PIVOTING do not.

We propose a simple method to solve this problem: we convert scores without probabilistic interpretation to ones that are scaled to the zero-one simplex. Given mention m_i and its top-n candidate entity list $\mathbf{E}_i$ along with their scores $\mathbf{S}_i$, the re-calibrated scores are identified as:

$$p_{i,j} = \frac{\exp(\gamma \times s_{i,j})}{\sum_{s_{i,k}\in\mathbf{S}_i} \exp(\gamma \times s_{i,k})} \quad (5)$$

where γ is a hyper-parameter that controls the peakiness of the distribution. After calibration, it is safe to combine prior scores with an average.

5.2 Feature Design

Next, we introduce the feature set for our disambiguation model, including features inspired by previous work (Sil and Florian, 2016; Ganea et al., 2016; Pan et al., 2017), as well as novel features specifically designed to tackle the low-resource scenario. We intentionally avoid features that take source language context words into consideration, as these would be heavily reliant on $\mathbb{W}_{\text{eng}}$ and $\mathbb{M}$ and weaken the transferability of the model. The formulation and resource requirements of unary and binary features are shown in the top and bottom halves of Table 1 respectively.

For unary features, we consider the number of mentions an entity is related to as f_l^3, where we consider the entity $e_{i,j}$ related to mention m_k if it co-occurs with any candidate entity of m_k (Moro et al., 2014). We also add the entity prior score f_l^2 among the whole Wikipedia (Yamada et al., 2017) to reflect the entity's overall salience. The exact match number f_l^4 indicates mention coreference.

For binary features, we attempt to deal with the noise and sparsity inherent in the co-occurrence counts of f_g^1. To tackle noise, we calculate the smoothed Positive Pointwise Mutual Information (PPMI) (Church and Hanks, 1990; Ganea et al., 2016) between two entities as f_g^2, which robustly estimates how much more the two entities co-occur than we expect by chance. To tackle sparsity, we incorporate English entity embeddings of Yamada et al. (2017), and calculate embedding similarity between two entities as f_g^3. Similar techniques have also been used by existing works (Ganea and Hofmann, 2017; Kolitsas et al., 2018). We also add the hyperlink count f_g^4 between a pair of entities as, if entity e_i's Wikipedia page mentions e_j, they are likely to be related.

We name our proposed feature set that includes all features listed in Table 1 as FEAT.

5.3 BURN: Feature Combination Model

With the growing number of features, we posit that a linear model with greedy entity pair selection (Section 3.2) is not expressive enough to take advantage of a rich feature set. Yamada et al. (2017) use Gradient Boosted Regression Trees (GBRT; Friedman (2001)) to combine features, but GBRTs do not allow for end-to-end training and thus constrain the flexibility of the model. Ganea et al. (2016); Ganea and Hofmann (2017) propose to use Loopy Belief Propagation (LBP; Murphy et al. (1999)) to estimate the global score (Equation (2)) and use non-linear functions to combine local and global scores (Equation (1)). However, BP is challenging to implement, and previous work has not attempted to combine more fine-grained features (e.g. unary feature $\mathbf{\Phi}(e_{i,j})$) non-linearly.

Instead, we propose a *belief update recurrent network* (BURN) that combines features in a non-linear and iterative fashion. Compared to existing work (Naradowsky and Riedel, 2016; Ganea et al., 2016; Ganea and Hofmann, 2017) as well as our base model, the advantages of BURN are: (1) it is easy to implement with existing neural network toolkits, (2) parameters can be learned end-to-end, (3) it considers non-linear combinations over more fine-grained features and thus has potential to fit more complex combination patterns, (4) it can model (distance) relations between mentions in the document.

Given unary feature vector $\mathbf{\Phi}(e_{i,j})$ with d_l features, BURN replaces the linear combination in Equation (1) with two fully connected layers:

$$s_l(e_{i,j}|D) = \mathbf{W}_l^{2^T}(\sigma(\mathbf{W}_l^{1^T}\mathbf{\Phi}(e_{i,j}))) + \mathbf{W}_l^{3^T}\mathbf{\Phi}(e_{i,j})$$

where $\mathbf{W}_l^1 \in \mathbb{R}^{d_l \times h_l}$, $\mathbf{W}_l^2 \in \mathbb{R}^{h_l \times 1}$ and $\mathbf{W}_l^3 \in \mathbb{R}^{d_l \times 1}$. σ is a non-linear function, for which we use leaky rectified linear units (Leaky ReLu; Maas et al. (2013)). We add a linear addition of the input to alleviate the gradient vanishing problem. Equation (4) is revised in a similar way.

As discussed in Equation (3), our baseline model calculates the mention evidence greedily. However, there may be many candidate entities for each mention, some containing noise. BURN solves this problem by weighting $s_e(e_{i,j}, e_{k,w})$ with the current entity probability $p(e_{k,w}|D)$. An illustration is in the bottom of Figure 2. The evidence from m_k is now defined as:

$$s_m(m_k, e_{i,j}) = \sum_{w=1}^{|C_k|} s_e(e_{i,j}, e_{k,w}) p(e_{k,w}|D) \quad (6)$$

Instead of simply averaging mention evidence in Equation (2), we also use a gating function to control the influence of m_k's mention evidence on m_i (top of Figure 2), giving score

$$s_g(e_{i,j}|D) = \sum_{k \neq i} g_m(m_i, m_k) s_m(m_k, e_{i,j})$$

The gating function g is essentially a lookup table that has one scalar for each distance (in words) between two mentions. We train this table along with all other parameters of the model. The motivation for this gating function is that a mention is more likely to be coherent with a nearby mention than a distant one. We assume that this is true for almost all languages, and thus will be useful even without training in the language to be processed.

As shown in Equation (6), there is a circular dependency between entities. To solve this problem, we iteratively update the probability of entities until convergence or reaching a maximum number of iterations T. In iteration t, the calculation of s_m will use entity probabilities from iteration $t-1$. The revised Equation (6) is as follows:

$$s_m^t(m_k, e_{i,j}) = \sum_{w=1}^{|C_k|} s_e(e_{i,j}, e_{k,w}) p^{t-1}(e_{k,w}|D)$$

Unrolling this network through iterations, we can see that this is in fact a recurrent neural network.

Training BURN: The weights of BURN are learned end-to-end with the objective function:

$$L(\mathcal{D}, \mathcal{E}) = -\sum_{D \in \mathcal{D}} \sum_{m_i \in D} \log(p^T(e_i^*|D)).$$

As discussed above, the disambiguation model is fully language-agnostic and it does not require any annotated EL data or other resources in the source language. The model weights W_l, W_g and the lookup table g_m of gating function are trained on the TAC-KBP 2010 English training set (Ji et al., 2010) *only* and used as-is in another language. We use TAC-KBP 2012 English test set (Mayfield and Javier, 2012) as our development set.

6 Experiment II: Improving Low-resource XEL

Section 4 demonstrated a dramatic performance degradation for XEL in realistic low-resource settings. In this section, we evaluate the utility of our proposed methods that improve low-resource XEL.

6.1 Training Details

All models are implemented in PyTorch (Paszke et al., 2017). The size of the pre-trained entity embeddings (Yamada et al., 2017) is 300, trained with a window size of 15 and 15 negative samples. The hidden size h of both $\mathbf{W_l^1}$ and $\mathbf{W_g^1}$ is set to 128, the dropout rate is set to 0.5. For the gating function, we set mention distances that are larger than 50 tokens to 50, then bin the distances with a bin size of 4. We only consider the 30 nearest context mentions for each mention. The maximum number of iterations for inference is set to 20. We use the Adam optimizer with the default learning rate (`1e-3`) to train the model. The γ of calibrated candidate combination is set to 1. It takes around two hours to train a GREEDY model and ten hours to train a BURN model with a Titan X GPU, regardless of the feature set.

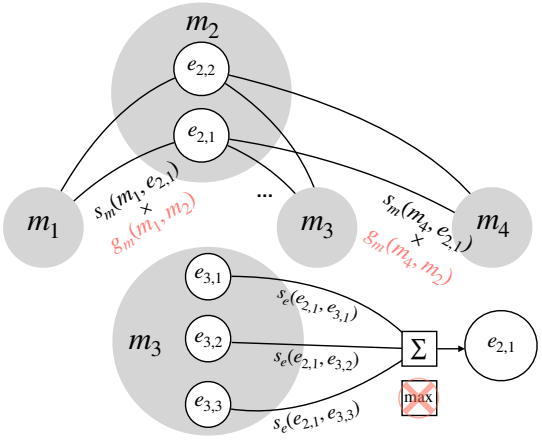

Figure 2: Top: the global score of an entity is a *weighted* aggregation of mention evidence from context mentions, instead of an average. Bottom: each mention evidence is a *weighted* entity-pair score, instead of the max.

6.2 Results

Table 3 compares models on the datasets we introduce in Section 4. Given that the critical issue was the degradation of candidate recall of the resource-heavy WIKIMENTION method in low-resource settings (Section 4), we first examine the alternative resource-light PIVOTING model. The first rows of block 1 and 2 of the table show the gold candidate recall of each method. While PIVOTING greatly exceeds WIKIMENTION on *ti*, which only has 168 Wikipedia pages, its performance is much lower on *si*, which has 15k pages. Overall, while these two models could outperform each other in their respective favorable settings (when a similar pivot language exists for the former, and when a large Wikipedia exists for the latter), it is challenging to decide which is more appropriate in the face of the realistic setting of existent, but scarce, resources.

Thus, in the third block of the table we show

Block Index	$\mathbb{W}_{\text{eng}}$	$\mathbb{W}_{\text{src}}$	$\mathbb{M}$	Candidates	Inference	ti	om	rw	si
1	✓			PIVOTING	Gold Candidate Recall	36.2	20.9	59.6	32.1
					$p(e\|m)$	32.9	18.2	54.9	11.8
					BASE + GREEDY	33.7	18.5	55.9	20.5
					FEAT + GREEDY	33.7	13.6	46.2	15.5
					BASE + BURN	**34.9**	**19.4**	**56.2**	**21.1**
					FEAT + BURN	34.5	17.8	50.9	10.6
2	✓	✓	✓	WIKIMENTION	Gold Candidate Recall	21.9	45.3	45.6	66.6
					$p(e\|m)$	21.5	**41.0**	45.1	63.1
					BASE + GREEDY	**21.8**	38.4	44.9	64.4
					FEAT + GREEDY	21.6	38.7	44.6	64.4
					BASE + BURN	**21.8**	39.9	44.3	**64.7**
					FEAT + BURN	**21.8**	39.9	**45.6**	**64.7**
3	✓	✓	✓	WIKIMENTION + PIVOTING	Gold Candidate Recall	38.3	62.0	69.4	75.2
					$p(e\|m)$	33.6	54.0	66.0	66.8
					BASE + GREEDY	34.4	53.3	67.3	68.1
					FEAT + GREEDY	34.5	50.3	57.8	67.2
					BASE + BURN	**35.6**	**54.5**	65.2	**70.3**
					FEAT + BURN	35.2	53.6	**67.5**	68.8

Table 3: Accuracy (%) of different systems. ✓shows the resource requirements. The performances of the end-to-end baseline system grayed. The performances of baseline disambiguation for each candidate generation model are underlined and numbers in **bold** show the best performance for each setting. $p(e|m)$ refers to the method that chooses the highest prior score provided by corresponding candidate generation method.

results for the hybrid candidate generation model which uses both WIKIMENTION and PIVOTING. Compared to WIKIMENTION, this method improves the gold candidate recall between 9 to 24% over all four low-resource languages. The improvement ($> 15\%$) is especially considerable for *om* and *rw*. This reflects the fact that there are a significant number of unique candidate entities retrieved by these two candidate generation methods, and developing a proper way to combine them together results in higher-quality candidate lists. Notably, this method has also increased the headroom for a disambiguation model to contribute – in contrast to the WIKIMENTION setting where the difference between prior $p(e|m)$ and gold accuracy was minimal, now there is a 3-9% accuracy gap between the two settings.

Next, we turn to methods that close this gap. Focusing on this third block of the table, we can see that the proposed disambiguation model can take advantage of better candidate lists and yields significantly better results on all four languages. Notably, we observe that BURN consistently yields the best performance over all languages, improving by 0.2 to 3.3% over GREEDY. This result demonstrates the advantage of iterative nonlinear feature combination in low-resource settings. In contrast, there is not a consistent improvement from the proposed feature set FEAT compared to the baseline BASE. This is interesting as FEAT+BURN outperformed BASE+BURN by more than 10% on the English development set on which it was validated. We suspect this is because the feature value distribution of the English training data is different from that of low-resource languages, leading to sub-optimal transfer. We leave training algorithms for bridging this gap as an interesting avenue of future work.

In the context of the end-to-end system, the combination of our proposed methods brings 6-23% improvement over the baseline system. For languages (*ti*, *om*, *rw*) where resources are relative scarce, the improvement is especially considerable, ranging from 13 to 23%, indicating that our work is a promising first step towards improving XEL in realistic low-resource scenarios.

7 Conclusion

This paper has made two major contributions to the study of low-resource cross-lingual entity linking (XEL). First, we perform an extensive empirical evaluation on the effect of different resource availability assumptions on XEL and demonstrate that (1) the accuracy of existing systems greatly degrades on true low-resource settings, and (2) standard candidate generation systems constrain the performance of end-to-end XEL. This fact has been under-discussed in existing work and we argue that more attention should be paid to candidate generation for low-resource XEL. Second, based

on our empirical study, we propose three methodologies for candidate generation and disambiguation that make the best use of limited resources we will have in realistic settings. Experimental results suggest that our proposed methodologies are effective under extremely limited-resource scenarios, giving improvements in 6-23% end-to-end linking accuracy over the baseline system.

An immediate future focus is further improving the performance of candidate generation models in realistic low-resource settings. Further, we could consider more sophisticated strategies for cross-lingual training of entity disambiguation systems that fill the gap between English training data and real world low-resource data.

8 Acknowledgements

We would like to thank the anonymous reviewers for their useful feedback. This material is based upon work supported in part by the Defense Advanced Research Projects Agency Information Innovation Office (I2O) Low Resource Languages for Emergent Incidents (LORELEI) program under Contract No. HR0011-15-C0114. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *11th conference of the European Chapter of the Association for Computational Linguistics*.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 277–285. Association for Computational Linguistics.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. 2016. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, pages 927–938. International World Wide Web Conferences Steering Committee.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629.

Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringaard, and Fernando Pereira. 2016. Collective entity resolution with multi-focal attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 621–631.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics.

Heng Ji, Ralph Grishman, and Hoa Dang. 2011. Overview of the TAC 2011 knowledge base population track. In *TAC 2011 Proceedings Papers*.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the TAC 2010 knowledge base population track. In *TAC 2010 Proceedings Papers*, volume 3, pages 3–3.

Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of TAC-KBP 2015 tri-lingual entity discovery and linking. In *TAC 2015 Proceedings Papers*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Xiao Ling, Sameer Singh, and Daniel S Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.

James Mayfield and Artiles Javier. 2012. Overview of the TAC 2012 knowledge base population track. In *TAC 2012 Proceedings Papers*.

Paul McNamee, James Mayfield, Dawn Lawrie, Douglas Oard, and David Doermann. 2011. Cross-language entity linking. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 255–263.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Kevin P Murphy, Yair Weiss, and Michael I Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc.

Jason Naradowsky and Sebastian Riedel. 2016. Represent, aggregate, and constrain: A novel architecture for machine reading from noisy sources. *arXiv preprint arXiv:1610.09722*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1946–1958.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

Shruti Rijhwani, Jiateng Xie, Graham Neubig, and Jaime Carbonell. 2019. Zero-shot neural transfer for cross-lingual entity linking. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, Honolulu, Hawaii.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Avirup Sil and Radu Florian. 2016. One for all: Towards language independent named entity linking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2255–2264.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual entity linking. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 589–598, San Diego, California. Association for Computational Linguistics.

Shyam Upadhyay, Nitish Gupta, and Dan Roth. 2018. Joint multilingual supervision for cross-lingual entity linking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2495.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2017. Learning distributed representations of texts and entities from knowledge base. *Transactions of the Association for Computational Linguistics*, 5:397–411.