Zero-shot Cross-Language Transfer of Monolingual Entity Linking Models

Elliot Schumacher **James Mayfield** **Mark Dredze**
Johns Hopkins University
eschuma7@jhu.edu mayfield@jhu.edu mdredze@cs.jhu.edu

Abstract

Most entity linking systems, whether mono or multilingual, link mentions to a single English knowledge base. Few have considered linking non-English text to a non-English KB, and therefore, transferring an English entity linking model to both a new document and KB language. We consider the task of zero-shot cross-language transfer of entity linking systems to a new language and KB. We find that a system trained with multilingual representations does reasonably well, and propose improvements to system training that lead to improved recall in most datasets, often matching the in-language performance. We further conduct a detailed evaluation to elucidate the challenges of this setting.

1 Introduction

Entity linking – the process of matching mentions of people, places or organizations with a relevant knowledge base (KB) entry – has often focused on linking English text. Cross-language linking often uses English KBs for matching to non-English text. While transferring a system to a new document language presents challenges, it does not consider issues that arise when transferring to a new KB language. KBs in different languages consider different topics, and matching text within the same language presents different challenges from building cross-language representations. People build KBs in many different languages, and we should explore how to link documents to these KBs.

This paper considers zero-shot cross-language adaptation of a trained entity linking system to a new monolingual setting: the same new language for both the query document and KB. We consider adaptation so as to utilize the extensive annotated data resources for English, improving entity linking on languages that have little to no training data. Consider the example in Figure 1, which links the Spanish language mention *Senado* (*Senate*) to the KB entry *Senado de la República* (*Senate of the Republic of Mexico*). An entity linker uses the mention text and surrounding sentence paired with the KB entry (including information such as the name, description) to score the likelihood of a match. Many approaches to entity linking learn these linkages by training on a set of hand-annotated links in the desired language. If there are no or few language-specific annotations, how can we train a model on an annotation-rich language to perform well on other languages?

Similar to the architecture used in a cross-language setting (Schumacher et al., 2021), we take a neural approach to entity linking and use a multilingual pretrained transformer model, XLM-Roberta (XLM-R) (Conneau et al., 2019), to build representations of the available text for a mention and candidate entity pair. We feed each of these representations through a feed forward neural model to produce a likelihood score. XLM-R is a multilingual model that yields robust representations of text in a wide variety of languages. However, we find that even with the cross-language ability of XLM-R, in-language annotation data is key to an accurate linker. We thus propose ways to improve zero-shot cross-language transfer of a trained linker from one language to another.

We adapt a method from Chen and Cardie (2018) to add an adversarial objective to linker training which uses an intermediate layer in the linker to transform language-specific embeddings to language-agnostic via a language classification module. Similar approaches (Chen et al., 2019) have been used in other multilingual NLP tasks, but have yet to be explored in EL. To train this language-agnostic layer, we force the language classifier alone to predict the incorrect language label for unannotated portions of the source (*e.g.,* English) and target (*e.g.,* Spanish) text. We jointly train the ranker and the language classifier using the correct source (*e.g.,* English) language labels.

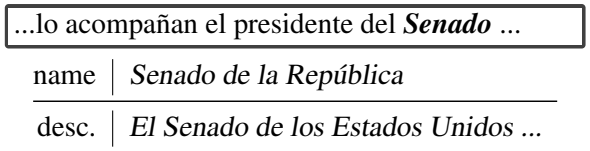

Figure 1: Example Spanish mention *Senado*, which is a link to the Spanish KB entity *Senado de la República* (the Senate of Mexico)

which encourages the name and mention representation to be language-independent.

Second, we augment the entity linker with information from the target language KB to capture popularity of each entity, better handling entities that are common in the target language but rare in the source. We find that both model adjustments improve zero-shot performance on several language pairs, and that the adversarial model specifically produces consistent improvement in recall. Overall, we demonstrate that entity linking models can be effectively adapted to a new language for both the query document and KB.

2 Entity Linking Model

Figure 1 shows an example mention in Spanish (*Senado*) linked to a Spanish-language KB entry – *Senado de la República* for the Spanish mention. A linker will compare the text of the mention to the name of the entity, and consider information available in the context of the mention (the surrounding sentences). the entity description, and the mention and entity types.

One approach to handling linking in multiple langages is to train separate models. While this works well for languages with a large amount of *annotated* data (English), others have far less (Spanish). Additionally, training a new model for each language does not scale well to many languages. Instead, we pursue building a model that can be trained on entity linking annotations in a single language and transferred to another without additional annotations: cross-language entity linking.

2.1 Architecture

We use a standard neural ranking architecture to focus on the mechanisms of transfer that has been applied successfully in cross-language entity linking (Schumacher et al., 2021). To score a mention m and and candidate entity e, we leverage a pointwise neural ranker inspired by the architecture of Dehghani et al. (2017). This produces a score for each mention-entity pair, creating a ranking of entities specific to each mention. Additionally, this pointwise approach allows scoring of previously unseen entities. We select a subset of entities to score using a triage system (§5.)

Our ranker captures two common sources of information about the entity – the mention string and entity name, and the context of the mention and the entity description. These sources are not KB specific (*e.g.,* type information) and thus transfer to different KBs. We create separate multilingual representations for the mention string and entity name (m_s and e_s), and the mention and entity context (m_c and e_c). The string and context pairs are fed into separate multilayer perceptrons (MLP), outputting an embedding that models the relationship between the entity and the mention. For example, we input m_s and e_s into a text-specific hidden layer h_s which outputs a combined representation r_s, and we input m_c and e_c into a context-specific hidden layer h_c which outputs a representation r_c. These representations r_s and r_c are then fed into a final MLP, which produces a score between -1 and 1.

To train our model parameters θ, we score a mention m and a correct entity link e_+, and separately score the same mention paired with n randomly sampled negative entities e_-. We apply hinge loss between the positive pair and the best performing negative pair;

$$L(\theta) = \mathbf{max}\{0, \epsilon - (S(\{m, e_+\}; \theta) - \mathbf{max}\{S(\{m, e_{0-}\}; \theta) \dots S(\{m, c_{n-}\}; \theta)\}\}$$

We use the resulting loss to backpropagate through the entire network. We use random combinations of parameters to select the best model configuration. For parameter values see Appendix Table 3.

2.2 Multilingual Representations

To create representations of the name and context for a mention-entity pair, we use XLM-Roberta (XLM-R) (Conneau et al., 2019), a multilingual transformer representation model. XLM-R outperforms other transformer models (such as mBERT (Devlin et al., 2019)) on multilingual tasks, and we confirmed this behavior in our initial experiments. Consider the Spanish example in Figure 1. We create a representation of the mention text m_s, *Senado*, by feeding the entire sentence through XLM-R, and form a single representation using max pooling on only the subwords of the

mention. We create a representation of the entity name e_s, *Senado de la República* in the same way, except without any surrounding context.

To create m_c, we select the sentences surrounding the mention up to XLM-R's sub-word limit. We use max pooling over XLM-R to create a single representation, following Schumacher et al. (2021). The same process is used to encode the entity context e_c, but uses the definition in the KB, using the first 512 subword tokens from that description.

3 Multilingual Transfer

The use of XLM-R makes our model inherently multilingual, allowing a single model to build representations in several languages. While this allows our models to do fairly well on previously unseen languages, we consider ways to further improve models during transfer: adaptation of the name matching model, and adaptation to the new knowledge base.

3.1 Language Adaptation

One source of error may arise from a linker learning language-specific patterns which do not generalize to other languages. Consider the example in Figure 1: would the model recognize that Spanish mention *Senado* is not linked to the *United States Senate*? While XLM-R provides a multilingual representation, the entity linking model has not been trained to learn this nuance in Spanish.

We add an adversarial objective to ensure that the model focuses on language-agnostic representations of the text, which will better transfer to other languages. The advantage of this approach is that it does not require annotated training data, but uses unannotated data to encourage desired model behavior. Chen and Cardie (2018) train a text classification system with an adversarial objective that forces the network to learn domain-invariant features. In addition to a standard text classifier that uses features from a shared and domain specific feature extractor, they add a domain discriminator which uses the shared feature extractor as input. They run two training passes: 1) a training pass for the entire network that uses the correct classification and domain labels; 2) an adversarially trained domain discriminator and only the shared feature extractor, which uses the inverse of domain labels as the target. Prediction only uses the standard classification output. This objective improves performance when classifying text from previously unseen domains. We use this approach to learn language-invariant representations for our linking task, so they can be transferred to a new languages using only source-language linking annotations.

Our proposed adversarial approach is described in Algorithm 1 and illustrated in Figure 2. For each epoch, we first adversarially train the language classifier. Using pairs of unannotated English $\mathbb{A}$ and L2 $\mathbb{B}$ text, we create representations in the same method as for m_s as described §2.2. Initially, we use randomly selected names from the ontology for $\mathbb{A}$ and $\mathbb{B}$ (see §6.3 for other approaches). Each of the two representations are fed into the shared invariant layer h_{s0}, the language classifier h_{adv}, and softmaxed to produce separate language likelihood scores for the English p_A and L2 p_B text. Importantly, we calculate the mean squared error (MSE) using the inverted language labels – for the English input, we calculate the error as if it was labelled as L2, and for the L2 input, we treat it as English. If we train with multiple L2 languages at the same time; all incorrect labels are applied with equal probability. We stop training the adversarial step after 50 epochs for one dataset (Wiki) based on development data performance.

We also run a standard entity linking training pass, in which we jointly train the linker and the language classifier using our set of training mentions $\mathbb{M}$ and corresponding entity labels $\mathbb{E}$. The entity linking loss is unchanged from §2.1, except that the m_s and e_s are first fed separately through the shared invariant layer h_{s0}. All h hidden layers in the model are randomly initialized weight vectors and learned in the training process. The loss for the language classifier is unchanged from the first step except that the correct labels are used. The effect of the language classifier loss is controlled by the parameter λ, which we set to be either 0.25 or 0.01 depending on the dataset. Models including this are referred to as **+A**. Further implementation details are available in §6.3. We experimented with adding the additional layers h_s0 and not applying the adversarial objective, and feeding both the language-invariant (*e.g.,* m) and language-specific representations (*e.g.,* r_m)) into the linker, but both performed worse in development experiments.

4 Algorithms

4.1 KB Adaptation

A second source of error comes from a change in the coverage of the KB, not necessarily due to the

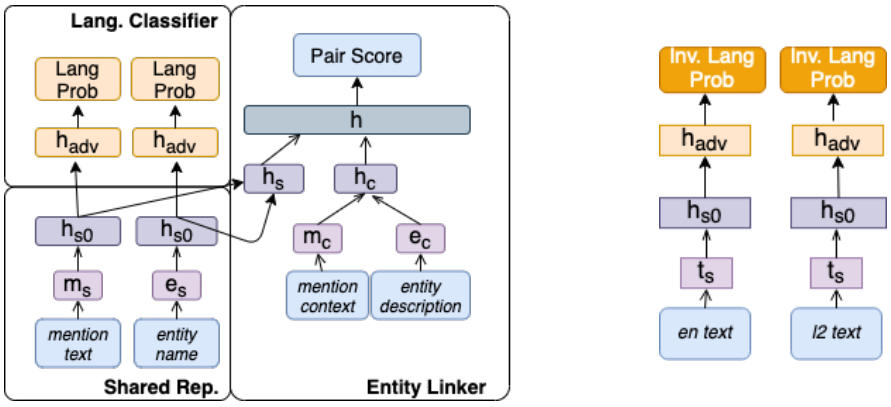

Figure 2: Our adversarial training approach consists of two steps – standard entity linking paired with training a language classifier (center), and adversarially training the language classifier (right). The hidden layer h_{s0} is shared.

Algorithm 1 Pseudo-code of adversarial model training. In each epoch, a random set of text ($y = 5$) is used to adversarially train the language classifier. Then, the entity linker and the language classifier with the correct labels are jointly trained.

Require: Mentions $\mathbb{M}$, entity labels $\mathbb{E}$; English Text $\mathbb{A}$; L2 Text $\mathbb{B}$; Hyperparameter $\lambda > 0$, y, $z \in N$, num_epochs

for $ep = 0$ to num_epochs **do**
  l_{adv}, l = 0
  for $i = 0$ to y **do** ▷ Adversarial Step
    t_A = representation of $\mathbb{A}_i$
    t_B = representation of $\mathbb{B}_i$
    $p_A = \mathcal{H}_{adv}(\mathcal{H}_{s0}(t_A))$
    $p_B = \mathcal{H}_{adv}(\mathcal{H}_{s0}(t_B))$ ▷ Calculate Lang scores
    l_{adv} += MSE(p_A, **L2**) + MSE(p_B, **ENG**) ▷ Calculate Loss using reversed labels
  Update $\mathcal{H}_{adv}$ using l_{adv}
  for $i = 0$ to z **do** ▷ Main Step
    m = representation of $\mathbb{M}_i$
    $r_m = \mathcal{H}_{s0}(m)$
    e = representation of $\mathbb{E}_i$
    $r_e = \mathcal{H}_{s0}(e)$
    l = EL Loss (Eq. 1) with r_m and r_e
    $p_M = \mathcal{H}_{adv}(r_m)$
    $p_E = \mathcal{H}_{adv}(r_e)$
    ▷ Calculate Lang scores
    l += λ (MSE(p_M, **ENG**) + MSE(p_E, **ENG**)) ▷ Calculate Loss using correct labels
  Update all parameters except $\mathcal{H}_{adv}$ using l

change in language. Trained entity linkers tend to do well on popular, or previously seen entities. New entities, which are common when a linker changes to a new KB, do worse. Consider the example in Figure 1: a linker trained on English will favor the KB entry for the U.S. Senate, more common in English language documents, as opposed to the Mexican Senate, which is more common in Spanish documents. This is especially important since we consider models transferred from TAC to our Wiki data (§5), which cover different topics.

We adapt the model to a KB in a new language by supplying the entity linker with popularity measures drawn from the new KB. This information could normally be derived from some annotated entity linking data, but in the zero-shot cross-language transfer setting we instead leverage the cross-links among entities in the KB, a good indicator of entity popularity. For example, the entity *Senado de la República* might have a link to the lower legislature of Mexico, *Cámara de Diputados*, and the President of Senate, *Presidente de la Cámara de Senadores*. Others, such as *Senado de Arizona*, are likely to have fewer. We count unique cross-links between entities, divide by the median number of links, and feed the result into the final feed forward neural network h (indicated as **+P**).

5 Datasets

We consider entity linking datasets in multiple languages from two sources. We treat each language as having a distinct KB, although entities may overlap in different languages. We predict NILs (mentions with no matching entity) as those where all candidate entities are below a given threshold (-1

unless otherwise noted). We evaluate using the script from Ji et al. (2015): Precision, Recall, F_1, and Micro-averaged precision. See Appendix Section A for implementation details.

TAC. The 2015 TAC KBP Entity Discovery and Linking dataset (Ji et al., 2015) consists of newswire and discussion posts in English, Spanish, and Mandarin Chinese. A mention is linked to NIL if there is no relevant entity in the KB. The KB is based on BaseKB. KB entities without non-English names are omitted.

Wiki. We created a multi-language entity linking dataset from Wikipedia links (Pan et al., 2017a) for Farsi and Russian. A preprocessed version of Wikipedia[1] is annotated with links to in-language pages, which we treat as entities. We consider this to be silver-standard data because–unlike TAC – the annotations are automatically derived. Thus the resulting distribution of mentions is different. Comparing the number of exact matches between the mention text and the entity name in Wikipedia (*e.g.,*, in Farsi 54.5%) to TAC (*e.g.,*, in Spanish 21.2%) underscores that TAC is a more illustrative dataset, thus we caution against treating Wikipedia as a replacement for a human-annotated entity linking dataset.

Triage. We use the triage system of Upadhyay et al. (2018), which retrieves a reduced set of entities for a mention for us to score. For a given gold mention m, a triage system will provide a set of k candidate entities $e_1 \dots e_k$. The system uses Wikipedia cross-links to generate a prior probability $\mathtt{P}_{\mathtt{prior}}(e_i|m)$ by estimating counts from those mentions. Originally, this system was designed to produce links for non-English mentions to English titles. We tweak this approach by applying the same pipeline, but for in-language titles, which did not require any major algorithmic adaptations.

6 Model Evaluation

We begin with a zero-shot evaluation: how well does a model trained on English (TAC) transfer to a new language without in-language training data? This baseline, which uses the same architecture as Schumacher et al. (2021), leverages only the crosslingual ability of XLM-R to apply English language annotations to the new languages. We evaluate the English trained model on Spanish (es) and Chinese (zh) for TAC, and Russian (ru) and Farsi (fa) for Wiki. We also train a separate model for each of these languages to establish an in-language performance baseline. We illustrate the difference in performance of an English-only model as compared to an in-language trained one in Figure 3; the dashed line above each metric shows the increase in performance. To control for the effect of training set size we ensure that the training sets are of equivalent size for each language by randomly downsizing the larger training dataset (*e.g.,* English) to match the smaller (*e.g.,* Spanish). For comparison, we include a simple nearest neighbor baseline (noted as **nn**), which selects the highest scoring mention-entity pair using cosine similarity between the mention name m_s and the entity representation e_s.

We then apply our language (noted as **+A**) and KB (noted as **+AP**) adaptation strategies for each language, and measure the performance on both the target and English language. In all cases, reported metrics are averaged over three runs. We report results for each language in the form of micro-averaged precision (micro), recall (r), and F_1. See Appendix Table 4 for full results and additional metrics, and Tables 5 and 6 for development results.

6.1 Transfer Performance

Figure 3 shows that zero-shot cross-language transfer from English gives worse performance compared to in-language models. Absolute values are included in Appendix Table 4. For TAC languages (es and zh) there is a large decrease in micro-avg and F_1, and the same for Wiki languages (fa and ru), except that F_1 decreases more significantly than recall, illustrating a drop in precision. The overall drop in performance is not large - the largest drop in F_1 is only .1 less compared to the in-language baseline. This illustrates that the linker is able to transfer across language and knowledge bases effectively. Compared to the baseline nearest neighbor model, which one has the higher performance improvement depends on the language. For example, while Spanish F_1 is nearly the same, Chinese F_1 is slightly higher with the **nn**, but in Farsi the English-trained model is an improvement for F_1.

We also evaluate other languages as sources of transfer. Appendix Table 4 shows results on training models on Chinese using the **+A** approach and testing on Spanish, demonstrating that our results are not specific to English. Note that the same

[1]We thank the authors of Pan et al. (2017a) for providing us with a preprocessed Wikipedia. We will work with the authors to release the dataset.

pattern appears when transferring from a Chinese trained model to a Spanish model. While the Spanish performance is understandably worse when transferring from Chinese instead of English, the reduction of F_1 performance is only –.086.

6.2 Language and KB adaptation

We train the TAC and Wiki datasets with different configurations based on development results (see §6.3): TAC: $\lambda = 0.25$ and the adversarial step covers all of training; Wiki: $\lambda = 0.01$ and stop the adversarial step after 50 epochs.

Applying the adversarial objective to English-trained models usually increases recall compared to the baseline English-trained models, and often even compared to the in-language trained models. For example, the English-trained, Chinese-tested model sees a large drop in recall which is almost completely eliminated when applying the adversarial objective. This increase in recall leads to nearly-equivalent F_1 performance in Spanish and Chinese in-language models and English trained models with the adversarial objective. In short, adversarial training greatly improves the models ability to locate the right KB entry, suggesting better name matching. This recall-focused improvement is useful for settings where high-recall is desired, such as in search. The exception to this is Farsi – this is likely because the high recall 0.934 of the zero-shot model established a high starting point. Compared to the nearest neighbor baseline, the **+A** outperforms the baseline in all languages for F_1, nn F_1, micro-avg., and recall. The same pattern appears when transferring a Chinese model instead of English. The F_1 performance is only –.017 below the in-language trained model despite not sharing a writing system.

We also explored transferring a multilingual model: training on English with **+A** and testing on all target languages at once (see Appendix Table 4). In almost all cases, the multilingual adversarial approach performs worse than a single-language one, but only slightly; it may be preferable when targeting multiple languages. KB popularity (**+AP**) has the largest effect on micro-average precision by doing much better on rarer entities, specifically in the TAC dataset. While in Chinese the improvement in micro-average is larger in the **+AP** models than in **+A**, in all other cases the micro-average is close to the **+A** model.

We explored model behavior on different types of entities using the TAC evaluation dataset and provided mention types (see Appendix Table A). For *Person* mentions, we see consistent performance between in-language, English, and English**+A** trained models. While this is not unexpected in Spanish (which has similar names to English), it is also true in Chinese, which uses a different orthography than English. The largest performance change occurred in *Geo-Political Entities*. For Chinese, F_1 drops 0.15 for an English trained model compared to an in-language trained model, but the deficit is erased in the English**+A** model. A similar pattern occurs in Spanish, suggesting that the adversarial model is able to improve the more challenging entity types.

6.3 Design of Adversarial objective

How does the configuration of the **+A** model change its behavior? We vary three factors and measure results on TAC evaluation (full results shown in Table 1): 1) the size of the coefficient λ; 2) whether to train using the entity linking objective only for an additional 50 epochs instead of for all epochs (for lower λ and additional entity linking training, we found that both worked better on Wiki development data, while a higher λ and full training worked better for TAC); and 3) training **+A** using randomly selected names from English and the target language plausibly learns a better name model than it does language-invariant representations, so we instead train with the first 512 subwords of randomly selected descriptions.

Comparing to a Chinese trained model, we considered versions with all non-baseline models trained on the joint entity linking and adversarial objective for 50 epochs, and the +EL models trained on EL data for an additional 50. Our reported setting for TAC, $\lambda = 0.25$ with name data, performs best on recall, F_1, and non-NIL F_1. However, when using the description data and $\lambda = 0.01$ with or without additional EL training, a better micro-averaged precision is achieved. Generally, the models using name data perform slightly better than those using descriptions, but the overall difference is slight (*e.g.,* +.009 F_1 for $\lambda = 0.25$ with name, –.015 F_1 with description), suggesting that the model is learning better multilingual representations. Finally, recall generally performs best with a higher λ and full adversarial training, and improves less with a lower λ and EL only training.

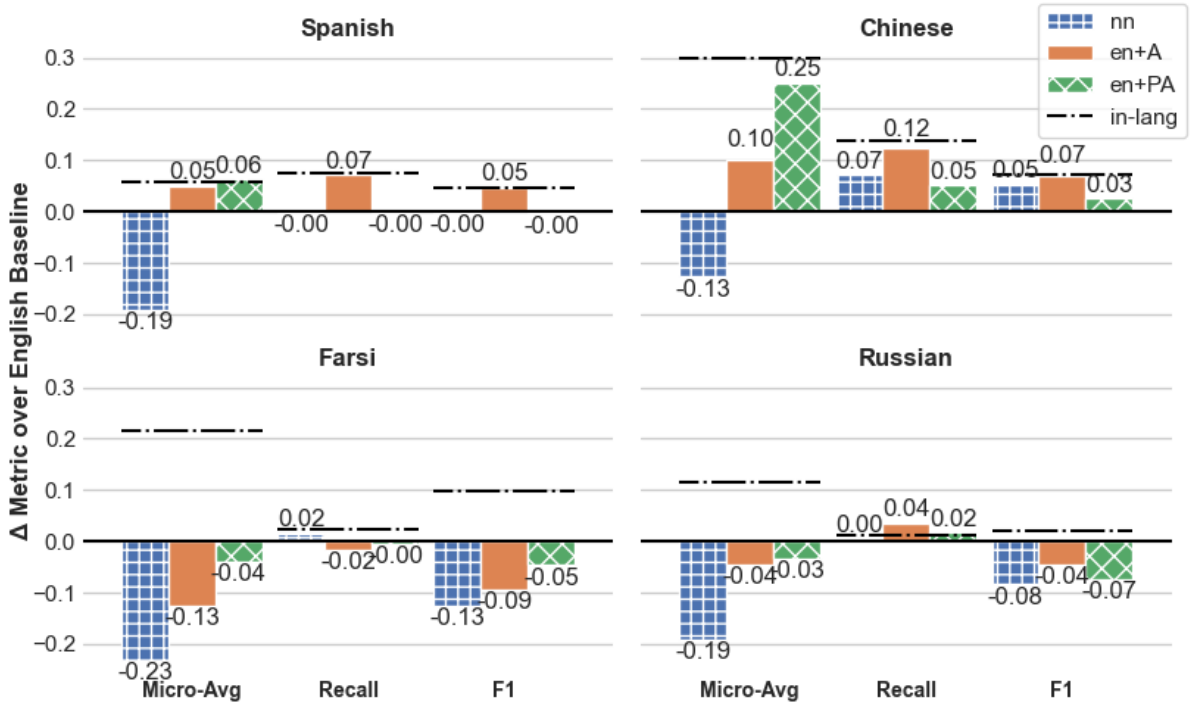

Figure 3: Compared to an English-only baseline (0.0 on y-axis), how do models with the adversarial objective (**+A**), the adversarial objective with popularity (**+PA**), and a nearest neighbor baseline (**nn**) perform? While in most cases, the performance of all models is below that of an in-language trained model (dashed line), **+A** most closely matches the recall in most cases. Additionally, **+PA** is best able to improve micro-average, especially compared to the poor performance of **nn**. All results and additional metrics are provided in Appendix Table 4.

6.4 Effect on English performance

What effect does forcing an English-trained model to better orient to a target language have on English-language performance? Table 2 shows TAC English evaluation results in three settings: 1) a baseline linker with English training data matched to the size of the target language's training data; 2) the added **+A** objective; 3) the added **+AP** objective. These are the same models as in Table 2, except tested on English.

Interestingly, the performance change is very small: a small increase for micro-average and a small decrease in F_1 and non-NIL F_1. The largest drop in performance is less than 0.05. This illustrates the capacity of the model: it can adapt to a new language while maintaining its performance on the source language.

6.5 Analysis

While our training methods are effective, they are inconsistent across our experiments. **+A** improves performance more on TAC data (Spanish and Chinese) than Wiki data (Farsi and Russian).

We postulate several explanations for this trend.

	Test	micro	r	F_1	nn F_1
	zh	0.674	0.789	0.824	0.846
	en base	−.341	−.123	−.060	−.071
+A name	.25	−.190	−.001	+.009	−.003
	.01	−.202	−.078	−.033	−.036
	.25+	−.205	−.123	−.062	−.073
	.01+	−.230	−.137	−.072	−.087
+A desc	.25	−.317	−.048	−.015	−.012
	.01	−.169	−.088	−.041	−.046
	.25+	−.287	−.188	−.108	−.133
	.01+	−.145	−.150	−.080	−.097

Table 1: How do adversarial settings affect performance? We consider the coefficient λ, type of text (names or descriptions), and entity-only training for 50 more epochs (*i.e.,* we stop updating the language classifier, indiated by +). Comparing an in-language to an English trained model using TAC Chinese evaluation, we find that $\lambda = .25$ with name data performs best in terms of recall, F_1, and nn F_1.

First, the distribution of mentions is different between the two datasets. The lexical similarity between mentions and entity names – one measure of how easy the mentions are to link – is much higher in Wiki. For Farsi development mentions, 54.5% were exact matches and also had an overall Jaro-Winkler (Winkler, 1990) lexical similarity of 94.1%. Compared to Spanish TAC (21.1% exact, 71.4% similarity) and Chinese (28% exact, 66.1% similarity), the Farsi data is relatively easy to link. While many entity linking studies rely on Wikipedia data due to its availability, it is not representative of other data types; we should build more human-annotated entity linking resources in non-English languages.

When comparing the drop in performance from an in-language trained model to an English trained model, recall drops in the TAC data, while precision drops in the Wiki data. The drop in precision may be due to the fact that we use English TAC data to train the zero-shot Wiki models, and that recall is fairly easy given the high mention-entity similarity. Another factor is the possibility that Wikipedia text is less suited as adversarial training data, compared to that from TAC. Thus, while see an increase in recall in the Wiki models, but this does not cancel out the reduction in precision.

7 Related Work

Many studies on entity linking (Dredze et al., 2010; Durrett and Klein, 2014; Gupta et al., 2017; Lample et al., 2016; Francis-Landau et al., 2016; Cao et al., 2018; Mueller and Durrett, 2018; Wang et al., 2015; Witten and Milne, 2008; Piccinno and Ferragina, 2014; Orr et al., 2020) have served as the basis for developing cross-language systems, as has increasing research in monolingual model transfer in other information extraction tasks (Johnson et al., 2019; Rahimi et al., 2019).

One multilingual model is Raiman and Raiman (2018), which transfers an English-trained system to French-language Wikipedia. They formulate a type system as a mixed integer problem, which they use to learn a type system from knowledge graph relations. Their training approach uses broad amounts of annotated data with type information (*e.g.,* all of English Wikipedia). Since we do not train English Wikipedia models, and also do not use that magnitude of training data, we were not able to produce numbers using their system that are comparable to ours despite our best efforts to do so.

Target	micro	F_1	nn F_1
en	0.484	0.672	0.797
zh+A	+.009	+.014	+.015
zh+P	+.030	−.025	−.031
en	0.472	0.678	0.802
es+A	+.004	−.014	−.017
es+P	+.011	−.036	−.043

Table 2: Compared to a baseline English TAC model (with training set size reduced to the noted language's training set size), we find that English performance is largely unchanged for both **+A** and **+P**.

Other recent work (Botha et al., 2020) uses a neural approach to link mentions in multiple languages, but differs from us by targeting language-agnostic KBs that include text in multiple languages. Work using unsupervised graph methods, such as Wang et al. (2015), are applied in non-English language pairs, such as Chinese, but are not transferred from a secondary language.

The related task of cross-language entity linking motivates approaches like transliteration (McNamee et al., 2011; Pan et al., 2017b), or monolingual entity linking paired with translation (Ji et al., 2015). Some (Tsai and Roth, 2016; Upadhyay et al., 2018) use the cross-language structure of Wikipedia to build entity linkers, or Rijhwani et al. (2019) study cross-language entity linking on low-resource languages.

8 Conclusion

We explored how to build a monolingually-trained entity linker that can be transferred to new languages that do not have annotated training data. With a neural ranker model using XLM-R, we see that while in-language trained models perform better than English-trained models applied to second languages, the performance decrease is not large.

We have validated several ways to improve these zero-shot models and find that an adversarial language classifier improves recall and F_1 on many datasets. Furthermore, by adjusting the adversarial parameters, different performance objectives can be achieved, such as maximizing recall. We also present an analysis of our models, demonstrating which settings have the highest expectation of success. Overall, we find that training the model to learn language-invariant representations is effective in improving performance when transferring to both text and a KB in a new language.

References

Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. Entity Linking in 100 Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, Online. Association for Computational Linguistics.

Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural collective entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 675–686, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2019. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112, Florence, Italy. Association for Computational Linguistics.

Xilun Chen and Claire Cardie. 2018. Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1226–1240, New Orleans, Louisiana. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Conference on Computational Linguistics (COLING)*, pages 277–285. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1256–1261, San Diego, California. Association for Computational Linguistics.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690, Copenhagen, Denmark. Association for Computational Linguistics.

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

Andrew Johnson, Penny Karanasou, Judith Gaspers, and Dietrich Klakow. 2019. Cross-lingual transfer learning for Japanese named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 182–189, Minneapolis, Minnesota. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

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, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

David Mueller and Greg Durrett. 2018. Effective use of context in noisy entity linking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1024–1029, Brussels, Belgium. Association for Computational Linguistics.

Laurel Orr, Megan Leszczynski, Simran Arora, Sen Wu, Neel Guha, Xiao Ling, and Christopher Re. 2020. Bootleg: Chasing the tail with self-supervised named entity disambiguation.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017a. 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.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017b. 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)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Francesco Piccinno and P. Ferragina. 2014. From tagme to wat: a new entity annotator. In *ERD '14*.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Jonathan Raphael Raiman and Olivier Michel Raiman. 2018. Deeptype: multilingual entity linking by neural type system evolution. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shruti Rijhwani, Jiateng Xie, Graham Neubig, and Jaime Carbonell. 2019. Zero-shot neural transfer for cross-lingual entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6924–6931.

Elliot Schumacher, James Mayfield, and Mark Dredze. 2021. Cross-lingual transfer in zero-shot cross-language entity linking. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 583–595, Online. Association for Computational Linguistics.

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, Stroudsburg, PA, USA. 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, Brussels, Belgium. Association for Computational Linguistics.

Han Wang, Jin Guang Zheng, Xiaogang Ma, Peter Fox, and Heng Ji. 2015. Language and domain independent entity linking with quantified collective validation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 695–704, Lisbon, Portugal. Association for Computational Linguistics.

William E Winkler. 1990. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *ERIC*.

Ian H Witten and David N Milne. 2008. An effective, low-cost measure of semantic relatedness obtained from wikipedia links.

A Dataset

TAC The training set consists of mentions across 447 documents, and the evaluation set consists of mention annotations across 502 documents. This leaves us $14,793$ development mentions, of which $11,344$ are non-NIL.

Wiki Some BaseKB entities used in the TAC dataset have Wikipedia links provided; we used those links as seed entities for retrieving mentions, retrieving a sample mention of those and adding the remaining links in the page. We mark 20% of the mentions as NIL.

Triage We use the system discussed in for both the **TAC** and **Wiki** datasets. However, while the triage system provides candidates in the same KB as the **Wiki** data, not all entities in the **TAC** KB have Wikipedia page titles. Therefore, the **TAC** triage step requires an intermediate step - using the Wikipedia titles generated by triage ($k = 10$), we query a Lucene database of BaseKB for relevant entities. For each title, we query BaseKB proportional to the prior provided by the triage system, meaning that we retrieve more BaseKB entities for titles that have a higher triage score, resulting in $l = 200$ entities. First, entities with Wikipedia titles are queried, followed by the entity name itself. If none are found, we query the mention string - this provides a small increase in triage recall.

Parameter	**Values**
Context Layer(s)	[768], **[512]**, [256], [512,256]
Mention Layer(s)	[768], **[512]**, [256], [512,256]
Final Layer(s)	**[512,256]**, [256,128], [128,64], [1024,512], [512], [256]
Dropout probability	0.1, **0.2**, 0.5
Learning rate	1e-5, 5e-4, **1e-4**, 5e-3, 1e-3

Table 3: To select parameters for the ranker, we tried 10 random combinations of the above parameters and selected the configuration that performed best on the TAC development set. The selected parameter is in bold.

	Spanish (es) evaluation					Chinese (zh) evaluation				
Training	micro	p	r	F_1	nn F_1	micro	p	r	F_1	nn F_1
same	0.623	0.910	0.711	0.798	0.870	0.670	0.862	0.787	0.822	0.844
nn	0.375	0.924	0.633	0.751	0.809	0.244	0.910	0.719	0.803	0.826
en	0.565	0.925	0.635	0.753	0.810	0.371	0.893	0.647	0.750	0.757
en+A	0.615	0.923	0.706	0.800	0.876	0.472	0.877	0.770	0.820	0.839
en+P	0.632	0.919	0.616	0.738	0.790	0.462	0.869	0.636	0.734	0.734
en+PA	0.628	0.921	0.633	0.750	0.808	0.622	0.871	0.698	0.775	0.790
en+A (all)	0.562	0.917	0.694	0.790	0.862	0.466	0.882	0.722	0.794	0.813
zh	0.492	0.924	0.579	0.712	0.755	—	—	—	—	—
zh+A	0.523	0.901	0.690	0.781	0.852	—	—	—	—	—
	Farsi (fa) evaluation					Russian (ru) evaluation				
Training	micro	p	r	F_1	nn F_1	micro	p	r	F_1	nn F_1
same	0.838	0.902	0.958	0.929	0.908	0.526	0.729	0.827	0.775	0.721
nn	0.392	0.560	0.950	0.705	0.585	0.362	0.654	0.868	0.746	0.680
en	0.623	0.748	0.934	0.830	0.774	0.552	0.798	0.863	0.829	0.791
en+A	0.498	0.616	0.918	0.737	0.639	0.508	0.697	0.899	0.785	0.729
en+A (all)	0.525	0.631	0.955	0.759	0.668	0.516	0.758	0.852	0.802	0.755
en+P	0.627	0.700	0.958	0.809	0.741	0.565	0.700	0.889	0.783	0.728
en+PA	0.584	0.679	0.930	0.785	0.709	0.519	0.661	0.881	0.755	0.691

Table 4: Compared to an in-language trained model and a nearest-neighbor baseline (**nn**), how does a zero-shot model trained only on English transfer? We find that while there is usually a performance improvement, it is often not large. Can we recover some of that lost performance by using an adversarial objective (**+A**) or adding knowledge base information (**+P**), or both (**+PA**)? We find that when applying an adversarial objective specifically, recall is increased leading to higher F_1 scores. For each setting, we report Micro-avg., precision, recall, F_1, and non-NIL F_1 on TAC and Wiki datasets.

Train /		All				Non-NIL				
Test	Model	micro	p	r	f1	micro	p	r	f1	Epoch
zh/zh	Baseline	0.795	0.890	0.830	0.859	0.801	0.884	0.884	0.884	50
en/zh	Baseline	0.202	0.905	0.697	0.788	0.077	0.899	0.721	0.800	100
en/zh	+A	0.439	0.897	0.732	0.806	0.367	0.892	0.764	0.823	50
en/zh	+A	0.381	0.911	0.756	0.827	0.296	0.907	0.794	0.847	50
en/zh	+PA	0.635	0.889	0.753	0.815	0.606	0.881	0.789	0.833	100
en/zh	+A (Desc)	0.266	0.908	0.718	0.802	0.156	0.903	0.747	0.818	
en/zh	+PA (Desc)	0.645	0.885	0.774	0.826	0.618	0.877	0.815	0.845	
en/zh	+P	0.544	0.894	0.685	0.776	0.494	0.888	0.707	0.787	200
es/es	Baseline	0.714	0.933	0.777	0.848	0.739	0.930	0.891	0.910	50
en/es	Baseline	0.488	0.942	0.643	0.764	0.444	0.944	0.716	0.815	100
en/es	+A	0.469	0.938	0.693	0.797	0.420	0.939	0.782	0.853	150
en/es	+A (multi)	0.548	0.952	0.753	0.841	0.523	0.956	0.860	0.906	50
en/es	+PA	0.654	0.931	0.695	0.796	0.660	0.931	0.784	0.851	100
en/es	+A (Desc)	0.496	0.943	0.737	0.828	0.455	0.949	0.839	0.891	
en/es	+PA (Desc)	0.650	0.937	0.692	0.796	0.656	0.939	0.780	0.852	
en/es	+P	0.664	0.928	0.698	0.797	0.674	0.930	0.788	0.853	150
zh/es	Baseline	0.378	0.942	0.661	0.777	0.301	0.943	0.739	0.829	550
zh/es	+A	0.514	0.939	0.785	0.855	0.479	0.945	0.902	0.923	49

Table 5: Single runs of Development TAC results for our reported models, and the training epoch we report for that configuration in the evaluation results table. Note that while we report results with the training sets equalized (zh and en training are set to be of equal size) for evaluation, the full development results do not have equalized training set sizes.

Train/Test	Model	micro	p	r	f1	Eval Epoch
ru/ru	Baseline	0.650	0.823	0.888	0.854	800
en/ru	Baseline	0.484	0.762	0.855	0.806	550
en/ru	+A	0.451	0.712	0.893	0.792	50
en/ru	+A (multi)	0.4188	0.6517	0.8652	0.7434	200
en/ru	+P	0.473	0.685	0.860	0.762	50
fa/fa	Baseline	0.832	0.881	0.966	0.922	800
en/fa	Baseline	0.603	0.720	0.928	0.811	150
en/fa	+A	0.447	0.555	0.948	0.700	200
en/fa	+A (multi)	0.448	0.538	0.966	0.691	50

Table 6: Single runs of Development Wiki results for select reported models, and the training epoch we report for that configuration in the evaluation results table. Note that while we report results with the training sets equalized (ru and en training are set to be of equal size) for evaluation, the full development results do not have equalized training set sizes. For the +AP model, we report at Epoch 150 for Russian and 200 for Farsi, and for +P Farsi we report Epoch 50 (same as in Russian). Note that with the Farsi +A (multi) model, since the best performing epoch was at 50, in effect to EL-only training was performed.

			In-Language			En			En+A		
type	lang	count	micro	r	f1	micro	r	f1	micro	r	f1
CMN	FAC	59	0.169	0.631	0.756	0.119	0.515	0.670	0.169	0.632	0.768
CMN	GPE	3933	0.856	0.906	0.912	0.108	0.685	0.796	0.510	0.887	0.916
CMN	LOC	461	0.729	0.947	0.886	0.488	0.810	0.840	0.547	0.933	0.892
CMN	ORG	1441	0.160	0.726	0.774	0.299	0.629	0.722	0.127	0.799	0.821
CMN	PER	3116	0.708	0.682	0.797	0.612	0.676	0.792	0.610	0.676	0.792
SPA	FAC	59	0.051	0.294	0.454	0.068	0.285	0.444	0.102	0.289	0.448
SPA	GPE	1570	0.664	0.891	0.927	0.338	0.674	0.791	0.532	0.830	0.888
SPA	LOC	174	0.144	0.824	0.874	0.672	0.717	0.810	0.787	0.863	0.892
SPA	ORG	799	0.451	0.681	0.782	0.444	0.678	0.779	0.444	0.691	0.788
SPA	PER	2022	0.715	0.624	0.755	0.693	0.602	0.741	0.723	0.624	0.755

Table 7: How do the results of in-language training compare to English-only trained models and models trained with the adversarial objective? We find that some types perform consistently, such as PER (or Persons) even in languages that do not share scripts. Others, such as GPE (Geo-Political Entities) and ORG (Organizations) see a substantial drop in performance when applying a English-only model, but see more of that regained when using an adversarial objective. These results are taken from a single run of the TAC evaluation data.