A Multi-lingual Multi-task Architecture for Low-resource Sequence Labeling

Ying Lin[1] *, **Shengqi Yang**[2], **Veselin Stoyanov**[3], **Heng Ji**[1]
[1] Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY, USA
{liny9,jih}@rpi.edu
[2] Intelligent Advertising Lab, JD.com, Santa Clara, CA, USA
sheqiyan@gmail.com
[3] Applied Machine Learning, Facebook, Menlo Park, CA, USA
vesko.st@gmail.com

Abstract

We propose a multi-lingual multi-task architecture to develop supervised models with a minimal amount of labeled data for sequence labeling. In this new architecture, we combine various transfer models using two layers of parameter sharing. On the first layer, we construct the basis of the architecture to provide universal word representation and feature extraction capability for all models. On the second level, we adopt different parameter sharing strategies for different transfer schemes. This architecture proves to be particularly effective for low-resource settings, when there are less than 200 training sentences for the target task. Using Name Tagging as a target task, our approach achieved 4.3%-50.5% absolute F-score gains compared to the mono-lingual single-task baseline model. [1]

1 Introduction

When we use supervised learning to solve Natural Language Processing (NLP) problems, we typically train an individual model for each task with task-specific labeled data. However, our target task may be intrinsically linked to other tasks. For example, Part-of-speech (POS) tagging and Name Tagging can both be considered as sequence labeling; Machine Translation (MT) and Abstractive Text Summarization both require the ability to understand the source text and generate natural language sentences. Therefore, it is valuable to transfer knowledge from related tasks to the target task. Multi-task Learning (MTL) is one of the most effective solutions for knowledge transfer across tasks. In the context of neural network architectures, we usually perform MTL by sharing parameters across models (Ruder, 2017).

Previous studies (Collobert and Weston, 2008; Dong et al., 2015; Luong et al., 2016; Liu et al., 2018; Yang et al., 2017) have proven that MTL is an effective approach to boost the performance of related tasks such as MT and parsing. However, most of these previous efforts focused on tasks and languages which have sufficient labeled data but hit a performance ceiling on each task alone. Most NLP tasks, including some well-studied ones such as POS tagging, still suffer from the lack of training data for many low-resource languages. According to Ethnologue[2], there are 7,099 living languages in the world. It is an unattainable goal to annotate data in all languages, especially for tasks with complicated annotation requirements. Furthermore, some special applications (e.g., disaster response and recovery) require rapid development of NLP systems for extremely low-resource languages. Therefore, in this paper, we concentrate on enhancing supervised models in low-resource settings by borrowing knowledge learned from related high-resource languages and tasks.

In (Yang et al., 2017), the authors simulated a low-resource setting for English and Spanish by downsampling the training data for the target task. However, for most low-resource languages, the data sparsity problem also lies in related tasks and languages. Under such circumstances, a single transfer model can only bring limited improvement. To tackle this issue, we propose a *multi-lingual multi-task* architecture which combines different transfer models within a unified architecture through two levels of parameter sharing. In the first level, we share character embeddings,

* Part of this work was done when the first author was on an internship at Facebook.

[1] The code of our model is available at https://github.com/limteng-rpi/mlmt

[2] https://www.ethnologue.com/guides/how-many-languages

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 799–809
Melbourne, Australia, July 15 - 20, 2018.

character-level convolutional neural networks, and word-level long-short term memory layer across all models. These components serve as a basis to connect multiple models and transfer universal knowledge among them. In the second level, we adopt different sharing strategies for different transfer schemes. For example, we use the same output layer for all Name Tagging tasks to share task-specific knowledge (e.g., I-PER[3] should not be assigned to the first word in a sentence).

To illustrate our idea, we take *sequence labeling* as a case study. In the NLP context, the goal of sequence labeling is to assign a categorical label (e.g., POS tag) to each token in a sentence. It underlies a range of fundamental NLP tasks, including POS Tagging, Name Tagging, and chunking.

Experiments show that our model can effectively transfer various types of knowledge from different auxiliary tasks and obtains up to 50.5% absolute F-score gains on Name Tagging compared to the mono-lingual single-task baseline. Additionally, our approach does not rely on a large amount of auxiliary task data to achieve the improvement. Using merely 1% auxiliary data, we already obtain up to 9.7% absolute gains in F-score.

2 Model

2.1 Basic Architecture

The goal of sequence labeling is to assign a categorical label to each token in a given sentence. Though traditional methods such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) (Lafferty et al., 2001; Ratinov and Roth, 2009; Passos et al., 2014) achieved high performance on sequence labeling tasks, they typically relied on hand-crafted features, therefore it is difficult to adapt them to new tasks or languages. To avoid task-specific engineering, (Collobert et al., 2011) proposed a feed-forward neural network model that only requires word embeddings trained on a large scale corpus as features. After that, several neural models based on the combination of long-short term memory (LSTM) and CRFs (Ma and Hovy, 2016; Lample et al., 2016; Chiu and Nichols, 2016) were proposed and achieved better performance on sequence labeling tasks.

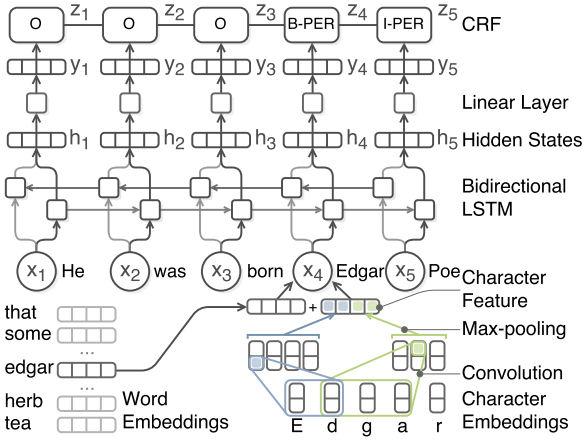

Figure 1: LSTM-CNNs: an LSTM-CRFs-based model for Sequence Labeling

LSTM-CRFs-based models are well-suited for multi-lingual multi-task learning for three reasons: (1) They learn features from word and character embeddings and therefore require little feature engineering; (2) As the input and output of each layer in a neural network are abstracted as vectors, it is fairly straightforward to share components between neural models; (3) Character embeddings can serve as a bridge to transfer morphological and semantic information between languages with identical or similar scripts, without requiring cross-lingual dictionaries or parallel sentences.

Therefore, we design our multi-task multi-lingual architecture based on the LSTM-CNNs model proposed in (Chiu and Nichols, 2016). The overall framework is illustrated in Figure 1. First, each word w_i is represented as the combination x_i of two parts, word embedding and character feature vector, which is extracted from character embeddings of the characters in w_i using convolutional neural networks (CharCNN). On top of that, a bidirectional LSTM processes the sequence $\boldsymbol{x} = \{x_1, x_2, ...\}$ in both directions and encodes each word and its context into a fixed-size vector h_i. Next, a linear layer converts h_i to a score vector y_i, in which each component represents the predicted score of a target tag. In order to model correlations between tags, a CRFs layer is added at the top to generate the best tagging path for the whole sequence. In the CRFs layer, given an input sentence $\boldsymbol{x}$ of length L and the output of the linear layer $\boldsymbol{y}$, the score of a sequence of tags $\boldsymbol{z}$ is

[3]We adopt the BIOES annotation scheme. Prefixes B-, I-, E-, and S- represent the beginning of a mention, inside of a mention, the end of a mention and a single-token mention respectively. The O tag is assigned to a word which is not part of any mention.

defined as:

$$S(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \sum_{t=1}^{L} (\boldsymbol{A}_{z_{t-1}, z_t} + \boldsymbol{y}_{t, z_t}),$$

where $\boldsymbol{A}$ is a transition matrix in which $\boldsymbol{A}_{p,q}$ represents the binary score of transitioning from tag p to tag q, and $\boldsymbol{y}_{t,z}$ represents the unary score of assigning tag z to the t-th word. Given the ground truth sequence of tags $\boldsymbol{z}$, we maximize the following objective function during the training phase:

$$\begin{aligned} \mathcal{O} &= \log P(\boldsymbol{z}|\boldsymbol{x}) \\ &= S(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) - \log \sum_{\tilde{\boldsymbol{z}} \in \boldsymbol{Z}} e^{S(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{z}})}, \end{aligned}$$

where $\boldsymbol{Z}$ is the set of all possible tagging paths.

We emphasize that our actual implementation differs slightly from the LSTM-CNNs model. We do not use additional word- and character-level explicit symbolic features (e.g., capitalization and lexicon) as they may require additional language-specific knowledge. Additionally, we transform character feature vectors using highway networks (Srivastava et al., 2015), which is reported to enhance the overall performance by (Kim et al., 2016) and (Liu et al., 2018). Highway networks is a type of neural network that can smoothly switch its behavior between transforming and carrying information.

2.2 Multi-task Multi-lingual Architecture

MTL can be employed to improve performance on multiple tasks at the same time, such as MT and parsing in (Luong et al., 2016). However, in our scenario, we only focused on enhancing the performance of a low-resource task, which is our target task or ***main task***. Our proposed architecture aims to transfer knowledge from a set of ***auxiliary tasks*** to the main task. For simplicity, we refer to a model of a main (auxiliary) task as a main (auxiliary) model.

To jointly train multiple models, we perform multi-task learning using parameter sharing. Let Θ_i be the set of parameters for model m_i and $\Theta_{i,j} = \Theta_i \cap \Theta_j$ be the shared parameters between m_i and m_j. When optimizing model m_i, we update Θ_i and hence $\Theta_{i,j}$. In this way, we can partially train model m_j as $\Theta_{i,j} \subseteq \Theta_j$. Previously, each MTL model generally uses a single transfer scheme. In order to merge different transfer models into a unified architecture, we employ two levels of parameter sharing as follows.

On the first level, we construct the basis of the architecture by sharing character embeddings, CharCNN and bidirectional LSTM among all models. This level of parameter sharing aims to provide universal word representation and feature extraction capability for all tasks and languages.

Character Embeddings and Character-level CNNs. Character features can represent morphological and semantic information; e.g., the English morpheme `dis-` usually indicates negation and reversal as in "*disagree*" and "*disapproval*". For low-resource languages lacking in data to suffice the training of high-quality word embeddings, character embeddings learned from other languages may provide crucial information for labeling, especially for rare and out-of-vocabulary words. Take the English word "*overflying*" (flying over) as an example. Even if it is rare or absent in the corpus, we can still infer the word meaning from its suffix `over-` (above), root `fly`, and prefix `-ing` (present participle form). In our architecture, we share character embeddings and the CharCNN between languages with identical or similar scripts to enhance word representation for low-resource languages.

Bidirectional LSTM. The bidirectional LSTM layer is essential to extract character, word, and contextual information from a sentence. However, with a large number of parameters, it cannot be fully trained only using the low-resource task data. To tackle this issue, we share the bidirectional LSTM layer across all models. Bear in mind that because our architecture does not require aligned cross-lingual word embeddings, sharing this layer across languages may confuse the model as it equally handles embeddings in different spaces. Nevertheless, under low-resource circumstances, data sparsity is the most critical factor that affects the performance.

On top of this basis, we adopt different parameter sharing strategies for different transfer schemes. For cross-task transfer, we use the same word embedding matrix across tasks so that they can mutually enhance word representations. For cross-lingual transfer, we share the linear layer and CRFs layer among languages to transfer task-specific knowledge, such as the transition score between two tags.

Word Embeddings. For most words, in addition to character embeddings, word embeddings are still crucial to represent semantic informa-

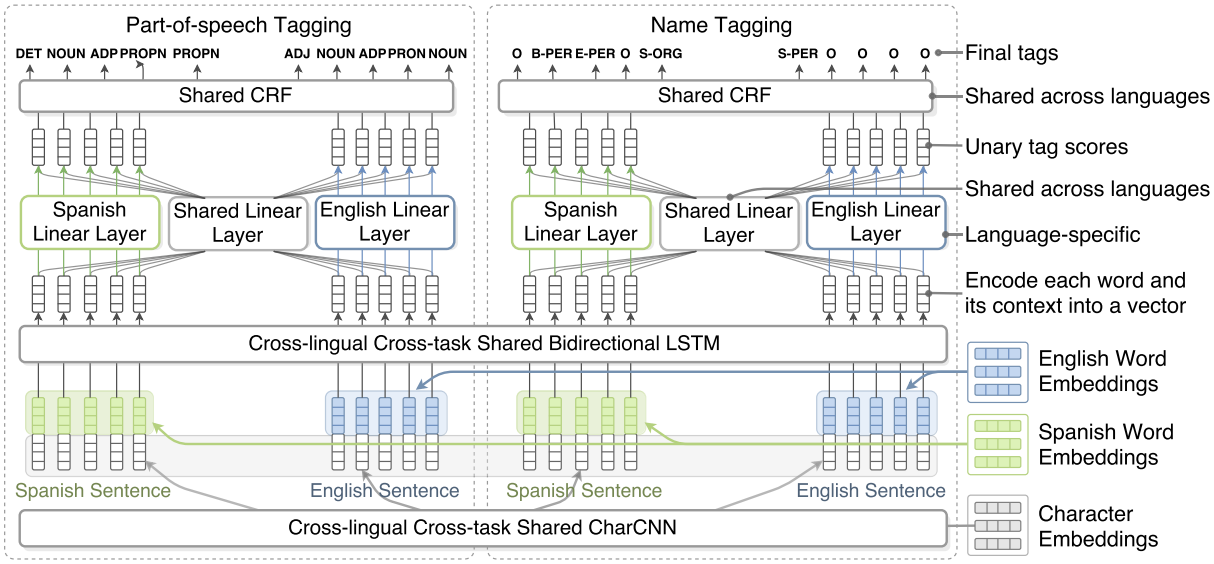

Figure 2: Multi-task Multi-lingual Architecture

tion. We use the same word embedding matrix for tasks in the same language. The matrix is initialized with pre-trained embeddings and optimized as parameters during training. Thus, task-specific knowledge can be encoded into the word embeddings by one task and subsequently utilized by another one. For a low-resource language even without sufficient raw text, we mix its data with a related high-resource language to train word embeddings. In this way, we merge both corpora and hence their vocabularies.

Recently, Conneau et al. (2017) proposed a domain-adversarial method to align two monolingual word embedding matrices without cross-lingual supervision such as a bilingual dictionary. Although cross-lingual word embeddings are not required, we evaluate our framework with aligned embeddings generated using this method. Experiment results show that the incorporation of cross-lingual embeddings substantially boosts the performance under low-resource settings.

Linear Layer and CRFs. As the tag set varies from task to task, the linear layer and CRFs can only be shared across languages. We share these layers to transfer task-specific knowledge to the main model. For example, our model corrects [S-PER Charles] [S-PER Picqué] to [B-PER Charles] [E-PER Picqué] because the CRFs layer fully trained on other languages assigns a low score to the rare transition S-PER→S-PER and promotes B-PER→E-PER. In addition to the shared linear layer, we add an unshared language-specific linear layer to allow the model to behave differently toward some features for different languages. For example, the suffix `-ment` usually indicates nouns in English whereas indicates adverbs in French.

We combine the output of the shared linear layer $\boldsymbol{y}^u$ and the output of the language-specific linear layer $\boldsymbol{y}^s$ using:

$$\boldsymbol{y} = \boldsymbol{g} \odot \boldsymbol{y}^s + (1 - \boldsymbol{g}) \odot \boldsymbol{y}^u,$$

where $\boldsymbol{g} = \sigma(\boldsymbol{W}_g \boldsymbol{h} + \boldsymbol{b}_g)$. $\boldsymbol{W}_g$ and $\boldsymbol{b}_g$ are optimized during training. $\boldsymbol{h}$ is the LSTM hidden states. As $\boldsymbol{W}_g$ is a square matrix, $\boldsymbol{y}$, $\boldsymbol{y}^s$, and $\boldsymbol{y}^u$ have the same dimension.

Although we only focus on sequence labeling in this work, our architecture can be adapted for many NLP tasks with slight modification. For example, for text classification tasks, we can take the last hidden state of the forward LSTM as the sentence representation and replace the CRFs layer with a Softmax layer.

In our model, each task has a separate object function. To optimize multiple tasks within one model, we adopt the alternating training approach in (Luong et al., 2016). At each training step, we sample a task d_i with probability $\frac{r_i}{\sum_j r_j}$, where r_i is the *mixing rate value* assigned to d_i. In our experiments, instead of tuning r_i, we estimate it by:

$$r_i = \mu_i \zeta_i \sqrt{N_i}\,,$$

where μ_i is the task coefficient, ζ_i is the language coefficient, and N_i is the number of training examples. μ_i (or ζ_i) takes the value 1 if the task

(or language) of d_i is the same as that of the target task; Otherwise it takes the value 0.1. For example, given English Name Tagging as the target task, the task coefficient μ and language coefficient ζ of Spanish Name Tagging are 0.1 and 1 respectively.

While assigning lower mixing rate values to auxiliary tasks, this formula also takes the amount of data into consideration. Thus, auxiliary tasks receive higher probabilities to reduce overfitting when we have a smaller amount of main task data.

3 Experiments

3.1 Data Sets

For Name Tagging, we use the following data sets: Dutch (NLD) and Spanish (ESP) data from the CoNLL 2002 shared task (Tjong Kim Sang, 2002), English (ENG) data from the CoNLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003), Russian (RUS) data from LDC2016E95 (Russian Representative Language Pack), and Chechen (CHE) data from TAC KBP 2017 10-Language EDL Pilot Evaluation Source Corpus[4]. We select Chechen as another target language in addition to Dutch and Spanish because it is a truly under-resourced language and its related language, Russian, also lacks NLP resources.

Code	Train	Dev	Test
NLD	202,931 (13,344)	37,761 (2,616)	68,994 (3,941)
ESP	207,484 (18,797)	51,645 (4,351)	52,098 (3,558)
ENG	204,567 (23,499)	51,578 (5,942)	46,666 (5,648)
RUS	66,333 (3,143)	8,819 (413)	7,771 (407)
CHE	98,355 (2,674)	12,265 (312)	11,933 (366)

Table 1: Name Tagging data set statistics: #token and #name (between parentheses).

For POS Tagging, we use English, Dutch, Spanish, and Russian data from the CoNLL 2017 shared task (Zeman et al., 2017; Nivre et al., 2017). In this data set, each token is annotated with two POS tags, UPOS (universal POS tag) and XPOS (language-specific POS tag). We use UPOS because it is consistent throughout all languages.

3.2 Experimental Setup

We use 50-dimensional pre-trained word embeddings and 50-dimensional randomly initialized character embeddings. We train word embeddings using the word2vec package[5]. English, Spanish, and Dutch embeddings are trained on corresponding Wikipedia articles (2017-12-20 dumps). Russian embeddings are trained on documents in LDC2016E95. Chechen embeddings are trained on documents in TAC KBP 2017 10-Language EDL Pilot Evaluation Source Corpus. To learn a mapping between mono-lingual word embeddings and obtain cross-lingual embeddings, we use the unsupervised model in the MUSE library[6] (Conneau et al., 2017). Although word embeddings are fine-tuned during training, we update the embedding matrix in a sparse way and thus do not have to update a large number of parameters.

We optimize parameters using Stochastic Gradient Descent with momentum, gradient clipping and exponential learning rate decay. At step t, the learning rate α_t is updated using $\alpha_t = \alpha_0 * \rho^{t/T}$, where α_0 is the initial learning rate, ρ is the decay rate, and T is the decay step.[7] To reduce overfitting, we apply Dropout (Srivastava et al., 2014) to the output of the LSTM layer.

We conduct hyper-parameter optimization by exploring the space of parameters shown in Table 2 using random search (Bergstra and Bengio, 2012). Due to time constraints, we only perform parameter sweeping on the Dutch Name Tagging task with 200 training examples. We select the set of parameters that achieves the best performance on the development set and apply it to all models.

Layer	Range	Final
CharCNN Filter Number	[10, 30]	20
Highway Layer Number	[1, 2]	2
Highway Activation Function	ReLU, SeLU	SeLU
LSTM Hidden State Size	[50, 200]	171
LSTM Dropout Rate	[0.3, 0.8]	0.6
Learning Rate	[0.01, 0.2]	0.02
Batch Size	[5, 25]	19

Table 2: Hyper-parameter search space.

3.3 Comparison of Different Models

In Figure 3, 4, and 5, we compare our model with the mono-lingual single-task LSTM-CNNs model (denoted as *baseline*), cross-task transfer model, and cross-lingual transfer model in low-resource settings with Dutch, Spanish, and Chechen Name Tagging as the main task respectively. We use English as the related language for Dutch and Spanish, and use Russian as the related language for

[4] https://tac.nist.gov/2017/KBP/data.html
[5] https://github.com/tmikolov/word2vec
[6] https://github.com/facebookresearch/MUSE
[7] Momentum β, gradient clipping threshold, ρ, and T are set to 0.9, 5.0, 0.9, and 10000 in the experiments.

Chechen. For cross-task transfer, we take POS Tagging as the auxiliary task. Because the CoNLL 2017 data does not include Chechen, we only use Russian POS Tagging and Russian Name Tagging as auxiliary tasks for Chechen Name Tagging.

We take Name Tagging as the target task for three reasons: (1) POS Tagging has a much lower requirement for the amount of training data. For example, using only 10 training sentences, our baseline model achieves 75.5% and 82.9% prediction accuracy on Dutch and Spanish; (2) Compared to POS Tagging, Name Tagging has been considered as a more challenging task; (3) Existing POS Tagging resources are relatively richer than Name Tagging ones; e.g., the CoNLL 2017 data set provides POS Tagging training data for 45 languages. Name Tagging also has a higher annotation cost as its annotation guidelines are usually more complicated.

We can see that our model substantially outperforms the mono-lingual single-task baseline model and obtains visible gains over single transfer models. When trained with less than 50 main tasks training sentences, cross-lingual transfer consistently surpasses cross-task transfer, which is not surprising because in the latter scheme, the linear layer and CRFs layer of the main model are not shared with other models and thus cannot be fully trained with little data.

Because there are only 20,400 sentences in Chechen documents, we also experiment with the data augmentation method described in Section 2.2 by training word embeddings on a mixture of Russian and Chechen data. This method yields additional 3.5%-10.0% absolute F-score gains. We also experiment with transferring from English to Chechen. Because Chechen uses Cyrillic alphabet , we convert its data set to Latin script. Surprisingly, although these two languages are not close, we get more improvement by using English as the auxiliary language.

In Table 3, we compare our model with state-of-the-art models using all Dutch or Spanish Name Tagging data. Results show that although we design this architecture for low-resource settings, it also achieves good performance in high-resource settings. In this experiment, with sufficient training data for the target task, we perform another round of parameter sweeping. We increase the embedding sizes and LSTM hidden state size to 100 and 225 respectively.

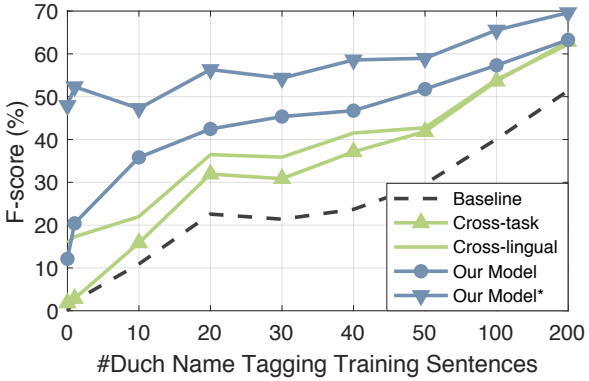

Figure 3: Performance on Dutch Name Tagging. We scale the horizontal axis to show more details under 100 sentences. Our Model*: our model with MUSE cross-lingual embeddings.

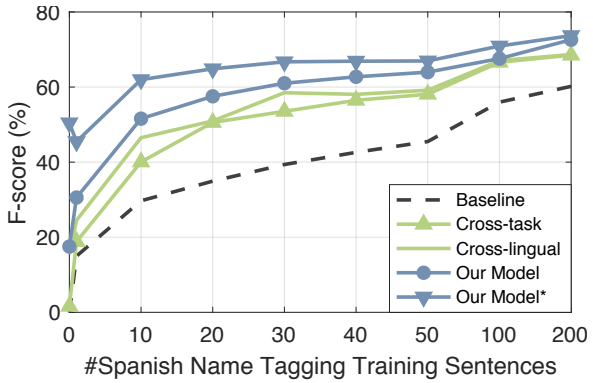

Figure 4: Performance on Spanish Name Tagging.

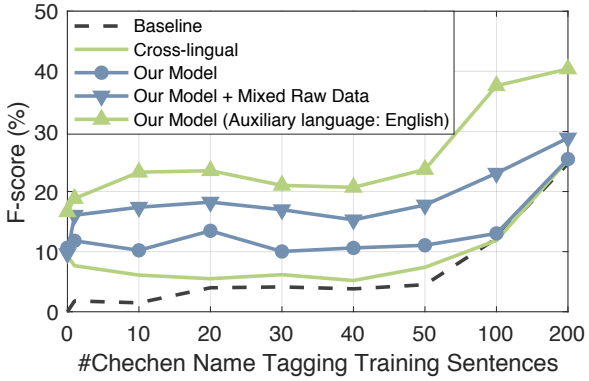

Figure 5: Performance on Chechen Name Tagging.

3.4 Qualitative Analysis

In Table 4, we compare Name Tagging results from the baseline model and our model, both trained with 100 main task sentences.

The first three examples show that shared character-level networks can transfer different levels of morphological and semantic information.

Language	Model	F-score
Dutch	Gillick et al. (2016)	82.84
	Lample et al. (2016)	81.74
	Yang et al. (2017)	85.19
	Baseline	85.14
	Cross-task	85.69
	Cross-lingual	85.71
	Our Model	**86.55**
Spanish	Gillick et al. (2016)	82.95
	Lample et al. (2016)	85.75
	Yang et al. (2017)	85.77
	Baseline	85.44
	Cross-task	85.37
	Cross-lingual	85.02
	Our Model	**85.88**

Table 3: Comparison with state-of-the-art models.

In example #1, the baseline model fails to identify "*Palestijnen*", an unseen word in the Dutch data, while our model can recognize it because the shared CharCNN represents it in a way similar to its corresponding English word "*Palestinians*", which occurs 20 times. In addition to mentions, the shared CharCNN can also improve representations of context words, such as "*staat*" (state) in the example. For some words dissimilar to corresponding English words, the CharCNN may enhance their word representations by transferring morpheme-level knowledge. For example, in sentence #2, our model is able to identify "*Rusland*" (Russia) as the suffix `-land` is usually associated with location names in the English data; e.g., Finland. Furthermore, the CharCNN is capable of capturing some word-level patterns, such as capitalized hyphenated compound and acronym as example #3 shows. In this sentence, neither "*PMS-centra*" nor "*MST*" can be found in auxiliary task data, while we observe a number of similar expressions, such as American-style and LDP.

The transferred knowledge also helps reduce overfitting. For example, in sentence #4, the baseline model mistakenly tags "*sección*" (section) and "*consellería*" (department) as organizations because their capitalized forms usually appear in Spanish organization names. With knowledge learned in auxiliary tasks that a lowercased word is rarely tagged as a proper noun, our model is able to avoid overfitting and correct these errors. Sentence #5 shows an opposite situation, where the capitalized word "*campesinos*" (farm worker) never appears in Spanish names.

In Table 5, we show differences between cross-lingual transfer and cross-task transfer. Although the cross-task transfer model recognizes "*Ingeborg Marx*" missed by the baseline model, it mistakenly assigns an S-PER tag to "*Marx*". Instead, from English Name Tagging, the cross-lingual transfer model borrows task-specific knowledge through the shared CRFs layer that (1) B-PER→S-PER is an invalid transition, and (2) even if we assign S-PER to "*Ingeborg*", it is rare to have continuous person names without any conjunction or punctuation. Thus, the cross-lingual model promotes the sequence B-PER→E-PER.

In Figure 6, we depict the change of tag distribution with the number of training sentences. When trained with less than 100 sentences, the baseline model only correctly predicts a few tags dominated by frequent types. By contrast, our model has a visibly higher recall and better predicts infrequent tags, which can be attributed to the implicit data augmentation and inductive bias introduced by MTL (Ruder, 2017). For example, if all location names in the Dutch training data are single-token ones, the baseline model will inevitably overfit to the tag S-LOC and possibly label "*Caldera de Taburiente*" as [S-LOC Caldera] [S-LOC de] [S-LOC Taburiente], whereas with the shared CRFs layer fully trained on English Name Tagging, our model prefers B-LOC→I-LOC→E-LOC, which receives a higher transition score.

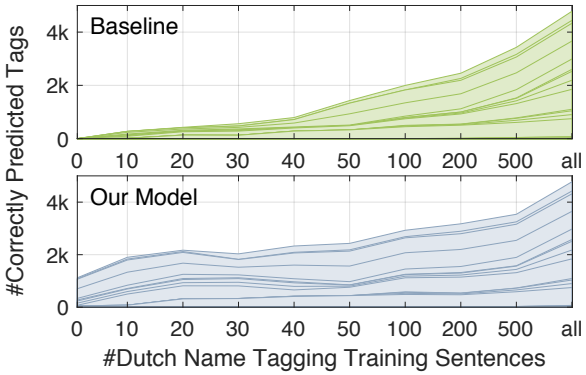

Figure 6: The distribution of correctly predicted tags on Dutch Name Tagging. The height of each stack indicates the number of a certain tag.

3.5 Ablation Studies

In order to quantify the contributions of individual components, we conduct ablation studies on Dutch Name Tagging with different numbers of training sentences for the target task. For the basic model, we we use separate LSTM layers and

#1 [DUTCH]: *If a Palestinian State is, however, the first thing the Palestinians will do.*
⋆ [B] Als er een Palestijnse staat komt, is dat echter het eerste wat de Palestijnen zullen doen
⋆ [A] Als er een [S-MISC Palestijnse] staat komt, is dat echter het eerste wat de [S-MISC Palestijnen] zullen doen
#2 [DUTCH]: *That also frustrates the Muscovites, who still live in the proud capital of Russia but can not look at the soaps that the stupid farmers can see on the outside.*
⋆ [B] Ook dat frustreert de Moskovieten , die toch in de fiere hoofdstad van Rusland wonen maar niet naar de soaps kunnen kijken die de domme boeren op de buiten wel kunnen zien
⋆ [A] Ook dat frustreert de [S-MISC Moskovieten] , die toch in de fiere hoofdstad van [S-LOC Rusland] wonen maar niet naar de soaps kunnen kijken die de domme boeren op de buiten wel kunnen zien
#3 [DUTCH]: *And the PMS centers are merging with the centers for school supervision, the MSTs.*
⋆ [B] En smelten de PMS-centra samen met de centra voor schooltoezicht, de MST's .
⋆ [A] En smelten de [S-MISC PMS-centra] samen met de centra voor schooltoezicht, de [S-MISC MST's] .
#4 [SPANISH]: *The trade union section of CC.OO. in the Department of Justice has today denounced more attacks of students to educators in centers dependent on this department ...*
⋆ [B] La [B-ORG sección] [I-ORG sindical] [I-ORG de] [S-ORG CC.OO.] en el [B-ORG Departamento] [I-ORG de] [E-ORG Justicia] ha denunciado hoy ms agresiones de alumnos a educadores en centros dependientes de esta [S-ORG consellería] ...
⋆ [A] La sección sindical de [S-ORG CC.OO.] en el [B-ORG Departamento] [I-ORG de] [E-ORG Justicia] ha denunciado hoy ms agresiones de alumnos a educadores en centros dependientes de esta consellería ...
#5 [SPANISH]: ... *and the Single Trade Union Confederation of Peasant Workers of Bolivia, agreed upon when the state of siege was ended last month.*
⋆ [B] ... y la [B-ORG Confederación] [I-ORG Sindical] [I-ORG Unica] [I-ORG de] [E-ORG Trabajadores] Campesinos de [S-ORG Bolivia] , pactadas cuando se dio fin al estado de sitio, el mes pasado .
⋆ [A] .. y la [B-ORG Confederación] [I-ORG Sindical] [I-ORG Unica] [I-ORG de] [I-ORG Trabajadores] [I-ORG Campesinos] [I-ORG de] [E-ORG Bolivia] , pactadas cuando se dio fin al estado de sitio, el mes pasado .

Table 4: Name Tagging results, each of which contains an English translation, result of the baseline model (B), and result of our model (A). The GREEN (RED) highlight indicates a correct (incorrect) tag.

[DUTCH] ... *Ingeborg Marx is her name, a formidable heavy weight to high above her head!*
⋆ [B] ... Zag ik zelfs onlangs niet dat een lief, mooi vrouwtje, Ingeborg Marx is haar naam, een formidabel zwaar gewicht tot hoog boven haar hoofd stak!
⋆ [CROSS-TASK] ... Zag ik zelfs onlangs niet dat een lief, mooi vrouwtje, [B-PER Ingeborg] [S-PER Marx] is haar naam, een formidabel zwaar gewicht tot hoog boven haar hoofd stak!
⋆ [CROSS-LINGUAL] ... Zag ik zelfs onlangs niet dat een lief, mooi vrouwtje, [B-PER Ingeborg] [E-PER Marx] is haar naam, een formidabel zwaar gewicht tot hoog boven haar hoofd stak!

Table 5: Comparing cross-task transfer and cross-lingual transfer on Dutch Name Tagging with 100 training sentences.

remove the character embeddings, highway networks, language-specific layer, and Dropout layer. As Table 6 shows, adding each component usually enhances the performance (F-score, %), while the impact also depends on the size of the target task data. For example, the language-specific layer slightly impairs the performance with only 10 training sentences. However, this is unsurprising as it introduces additional parameters that are only trained by the target task data.

Model	0	10	100	200	All
Basic	2.06	20.03	47.98	51.52	77.63
+C	1.69	24.22	48.53	56.26	83.38
+CL	9.62	25.97	49.54	56.29	83.37
+CLS	3.21	25.43	50.67	56.34	84.02
+CLSH	7.70	30.48	53.73	58.09	84.68
+CLSHD	12.12	35.82	57.33	63.27	86.00

Table 6: Performance comparison between models with different components (C: character embedding; L: shared LSTM; S: language-specific layer; H: highway networks; D: dropout).

3.6 Effect of the Amount of Auxiliary Task Data

For many low-resource languages, their related languages are also low-resource. To evaluate our model's sensitivity to the amount of auxiliary task data, we fix the size of main task data and down-sample all auxiliary task data with sample rates from 1% to 50%. As Figure 7 shows, the performance goes up when we raise the sample rate from

1% to 20%. However, we do not observe significant improvement when we further increase the sample rate. By comparing scores in Figure 3 and Figure 7, we can see that using only 1% auxiliary data, our model already obtains 3.7%-9.7% absolute F-score gains. Due to space limitations, we only show curves for Dutch Name Tagging, while we observe similar results on other tasks. Therefore, we may conclude that our model does not heavily rely on the amount of auxiliary task data.

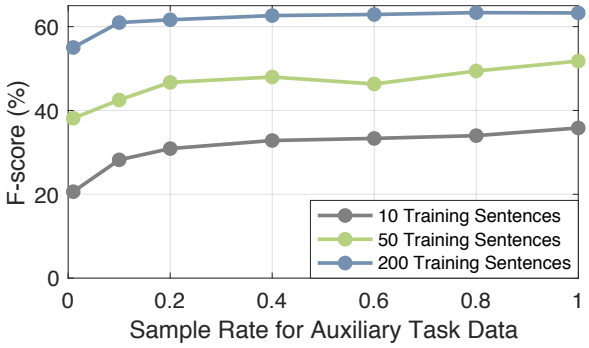

Figure 7: The effect of the amount of auxiliary task data on Dutch Name Tagging.

4 Related Work

Multi-task Learning has been applied in different NLP areas, such as machine translation (Luong et al., 2016; Dong et al., 2015; Domhan and Hieber, 2017), text classification (Liu et al., 2017), dependency parsing (Peng et al., 2017), textual entailment (Hashimoto et al., 2017), text summarization (Isonuma et al., 2017) and sequence labeling (Collobert and Weston, 2008; Søgaard and Goldberg, 2016; Rei, 2017; Peng and Dredze, 2017; Yang et al., 2017; von Däniken and Cieliebak, 2017; Aguilar et al., 2017; Liu et al., 2018)

Collobert and Weston (2008) is an early attempt that applies MTL to sequence labeling. The authors train a CNN model jointly on POS Tagging, Semantic Role Labeling, Name Tagging, chunking, and language modeling using parameter sharing. Instead of using other sequence labeling tasks, Rei (2017) and Liu et al. (2018) take language modeling as the secondary training objective to extract semantic and syntactic knowledge from large scale raw text without additional supervision. In (Yang et al., 2017), the authors propose three transfer models for cross-domain, cross-application, and cross-lingual transfer for sequence labeling, and also simulate a low-resource setting by downsampling the training data. By contrast, we combine cross-task transfer and cross-lingual transfer within a unified architecture to transfer different types of knowledge from multiple auxiliary tasks simultaneously. In addition, because our model is designed for low-resource settings, we share components among models in a different way (e.g., the LSTM layer is shared across all models). Differing from most MTL models, which perform supervisions for all tasks on the outermost layer, (Søgaard and Goldberg, 2016) proposes an MTL model which supervised tasks at different levels. It shows that supervising low-level tasks such as POS Tagging at lower layer obtains better performance.

5 Conclusions and Future Work

We design a multi-lingual multi-task architecture for low-resource settings. We evaluate the model on sequence labeling tasks with three language pairs. Experiments show that our model can effectively transfer different types of knowledge to improve the main model. It substantially outperforms the mono-lingual single-task baseline model, cross-lingual transfer model, and cross-task transfer model.

The next step of this research is to apply this architecture to other types of tasks, such as Event Extract and Semantic Role Labeling that involve structure prediction. We also plan to explore the possibility of integrating incremental learning into this architecture to adapt a trained model for new tasks rapidly.

Acknowledgments

This work was supported by the U.S. DARPA LORELEI Program No. HR0011-15-C-0115 and U.S. ARL NS-CTA No. W911NF-09-2-0053. 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

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López Monroy, and Thamar Solorio. 2017. A multi-task

approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *TACL*, 4:357–370.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Pius von Däniken and Mark Cieliebak. 2017. Transfer learning and sentence level features for named entity recognition on tweets. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *EMNLP*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *ACL*.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *NAACL HLT*.

Kazuma Hashimoto, Yoshimasa Tsuruoka, Richard Socher, et al. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *EMNLP*.

Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. 2017. Extractive summarization using multi-task learning with document classification. In *EMNLP*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT*.

Liyuan Liu, Jingbo Shang, Frank Xu, Xiang Ren, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *AAAI*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *ACL*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *ICLR*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *ACL*.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Aljoscha Burchardt, Marie Candito, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Silvie Cinková, Çağrı Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Tomaž Erjavec, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, John Lee, Phng Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Kaili Müürisep, Pinkey Nainwani, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lng

Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Vitaly Nikolaev, Hanna Nurmi, Stina Ojala, Petya Osenova, Robert Östling, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Larissa Rinaldi, Laura Rituma, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Benoît Sagot, Shadi Saleh, Tanja Samardžić, Manuela Sanguinetti, Baiba Saulīte, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Jonathan North Washington, Mats Wirén, Tak-sum Wong, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2017. Universal dependencies 2.1. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *CoNLL*.

Hao Peng, Sam Thomson, and Noah A Smith. 2017. Deep multitask learning for semantic dependency parsing. In *ACL*.

Nanyun Peng and Mark Dredze. 2017. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *ACL*.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *ACL*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *ICML*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *NAACL HLT*.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In *ICLR*.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, et al. 2017. CoNLL 2017 shared task: multilingual parsing from raw text to universal dependencies. In *CoNLL*.