Label-Agnostic Sequence Labeling by Copying Nearest Neighbors

Sam Wiseman **Karl Stratos**
Toyota Technological Institute at Chicago
Chicago, IL, USA
{swiseman,stratos}@ttic.edu

Abstract

Retrieve-and-edit based approaches to structured prediction, where structures associated with retrieved neighbors are edited to form new structures, have recently attracted increased interest. However, much recent work merely conditions on retrieved structures (e.g., in a sequence-to-sequence framework), rather than explicitly manipulating them. We show we can perform accurate sequence labeling by explicitly (and only) copying labels from retrieved neighbors. Moreover, because this copying is label-agnostic, we can achieve impressive performance in zero-shot sequence-labeling tasks. We additionally consider a dynamic programming approach to sequence labeling in the presence of retrieved neighbors, which allows for controlling the number of distinct (copied) segments used to form a prediction, and leads to both more interpretable and accurate predictions.

1 Introduction

Retrieve-and-edit style structured prediction, where a model retrieves a set of labeled nearest neighbors from the training data and conditions on them to generate the target structure, is a promising approach that has recently received renewed interest (Hashimoto et al., 2018; Guu et al., 2018; Gu et al., 2018; Weston et al., 2018). This approach captures the intuition that while generating a highly complex structure from scratch may be difficult, editing a sufficiently similar structure or set of structures may be easier.

Recent work in this area primarily uses the nearest neighbors and their labels simply as an additional context for a sequence-to-sequence style model to condition on. While effective, these models may not explicitly capture the discrete operations (like copying) that allow for the neighbors to be edited into the target structure, making interpreting the behavior of the model difficult. Moreover, since many retrieve-and-edit style models condition on dataset-specific labels directly, they may not easily allow for transfer learning and in particular to porting a trained model to a new task with different labels.

We address these limitations in the context of sequence labeling by developing a simple label-agnostic model that explicitly models copying token-level labels from retrieved neighbors. Since the model is not a function of the labels themselves but only of a learned notion of similarity between an input and retrieved neighbor inputs, it can be effortlessly ported (zero shot) to a task with different labels, without any retraining. Such a model can also take advantage of recent advances in representation learning, such as BERT (Devlin et al., 2018), in defining this similarity.

We evaluate the proposed approach on standard sequence labeling tasks, and show it is competitive with label-dependent approaches when trained on the same data, but substantially outperforms strong baselines when it comes to zero-shot transfer applications, such as when training with coarse labels and testing with fine-grained labels.

Finally, we propose a dynamic programming based approach to sequence labeling in the presence of retrieved neighbors, which allows for trading off token-level prediction confidence with trying to minimize the number of distinct *segments* in the overall prediction that are taken from neighbors. We find that such an approach allows us to both increase the interpretability of our predictions as well as their accuracy.

2 Related Work

Nearest neighbor based structured prediction (also referred to as instance- or memory-based learning) has a long history in machine learning and NLP,

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5363–5369
Florence, Italy, July 28 - August 2, 2019.

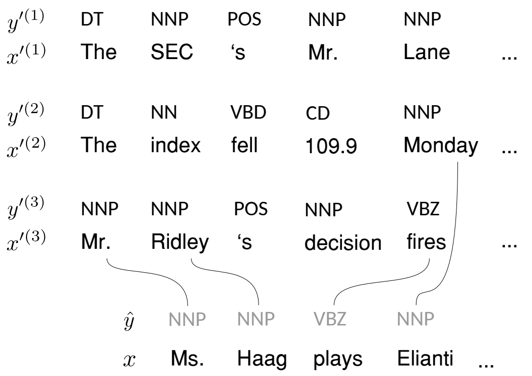

Figure 1: A visualization of POS tagging an input sentence x by copying token-labels from the label sequences $y'^{(m)}$ of $M = 3$ retrieved sentences $x'^{(m)}$.

with early successes dating back at least to the taggers of Daelemans (Daelemans, 1993; Daelemans et al., 1996) and the syntactic disambiguation system of Cardie (1994). Similarly motivated approaches remain popular for computer vision tasks, especially when it is impractical to learn a parametric labeling function (Shakhnarovich et al., 2006; Schroff et al., 2015).

More recently, there has been renewed interest in explicitly conditioning structured predictions on retrieved neighbors, especially in the context of language generation (Hashimoto et al., 2018; Guu et al., 2018; Gu et al., 2018; Weston et al., 2018), although much of this work uses neighbors as extra conditioning information within a sequence-to-sequence framework (Sutskever et al., 2014), rather than making discrete edits to neighbors in forming new predictions.

Retrieval-based approaches to structured prediction appear particularly compelling now with the recent successes in contextualized word embedding (McCann et al., 2017; Peters et al., 2018; Radford et al.; Devlin et al., 2018), which should allow for expressive representations of sentences and phrases, which in turn allow for better retrieval of neighbors for structured prediction.

Finally, we note that there is a long history of transfer-learning based approaches to sequence labeling (Ando and Zhang, 2005; Daume III, 2007; Schnabel and Schütze, 2014; Zirikly and Hagiwara, 2015; Peng and Dredze, 2016; Yang et al., 2017; Rodriguez et al., 2018, *inter alia*), though it is generally not zero-shot. There has, however, been recent work in zero-shot transfer for sequence labeling problems with binary token-labels (Rei and Søgaard, 2018).

3 Nearest Neighbor Based Labeling

While nearest-neighbor style approaches are compelling for many structured prediction problems, we will limit ourselves here to sequence-labeling problems, such as part-of-speech (POS) tagging or named-entity recognition (NER), where we are given a T-length sequence $x = x_{1:T}$ (which we will assume to be a sentence), and we must predict a corresponding T-length sequence of labels $\hat{y} = \hat{y}_{1:T}$ for x. We will assume that for any given task there are Z distinct labels, and denote x's true but unknown labeling as $y = y_{1:T} \in \{1, \dots, Z\}^T$.

Sequence-labeling is particularly convenient for nearest-neighbor based approaches, since a prediction $\hat{y}$ can be formed by simply concatenating labels extracted from the label-sequences associated with neighbors. In particular, we will assume we have access to a database $\mathcal{D} = \{x'^{(m)}, y'^{(m)}\}_{m=1}^{M}$ of M retrieved sentences $x'^{(m)}$ and their corresponding true label-sequences $y'^{(m)}$. We will predict a labeling $\hat{y}$ for x by considering each token x_t, selecting a labeled token $x_k'^{(m)}$ from $\mathcal{D}$, and then setting $\hat{y}_t = y_k'^{(m)}$.[1]

3.1 A Token-Level Model

We consider a very simple token-level model for this label-agnostic copying, where the probability that x's t'th label y_t is equal to $y_k'^{(m)}$ — the k'th label token of sequence $x'^{(m)}$ — simply depends on the similarity between x_t and $x_k'^{(m)}$, and is independent of the surrounding labels, conditioned on x and $\mathcal{D}$.[2] In particular, we define

$$p(y_t = y_k'^{(m)} \mid x, \mathcal{D}) \propto \exp(\mathbf{x}_t^{\mathsf{T}} \mathbf{x}_k'^{(m)}), \quad (1)$$

where the above probability is normalized over all label tokens of all label-sequences in $\mathcal{D}$. Above, $\mathbf{x}_t$ and $\mathbf{x}_k'^{(m)}$ (both in $\mathbb{R}^D$) represent the contextual word embeddings of the t'th token in x and the k'th token in $x'^{(m)}$, respectively, as obtained by running a deep sequence-model over x and over $x'^{(m)}$. In all experiments we use BERT (Devlin et al., 2018), a model based on the Transformer

[1] More precisely, we will set $\hat{y}_t$ to be an instance of the label *type* of which $y_k'^{(m)}$ is a label token; this distinction between label types and tokens can make the exposition unnecessarily obscure, and so we avoid it when possible.

[2] While recent sequence labeling models (Ma and Hovy, 2016; Lample et al., 2016), often model inter-label dependence with a first-order CRF (Lafferty et al., 2001), Devlin et al. (2018) have recently shown that excellent performance can be obtained by modeling labels as conditionally independent given a sufficiently expressive representation of x.

architecture (Vaswani et al., 2017), to obtain contextual word embeddings.

We fine-tune these contextual word embeddings by maximizing a latent-variable style probabilistic objective

$$\sum_{t=1}^{T} \ln \sum_{m=1}^{M} \sum_{k:\, y_k'^{(m)} = y_t} p(y_t = y_k'^{(m)} \,|\, x, \mathcal{D}), \quad (2)$$

where we sum over all individual label tokens in $\mathcal{D}$ that match y_t.

At test time, we predict $\hat{y}_t$ to be the label *type* with maximal marginal probability. That is, we set $\hat{y}_t$ to be $\arg\max_z \sum_{m=1}^{M} \sum_{k:\, y_k'^{(m)}=z} p(y_t = y_k'^{(m)} \,|\, x, \mathcal{D})$, where z ranges over the label types (e.g., POS or named entity tags) present in $\mathcal{D}$. As noted in the introduction, predicting labels in this way allows for the prediction of any label type present in the database $\mathcal{D}$ used at test time, and so we can easily predict label types unseen at training time without any additional retraining.

4 Data and Methods

Our main experiments seek to determine both whether the label-agnostic copy-based approach introduced above results in competitive sequence-labeling performance on standard metrics, as well as whether this approach gives rise to better zero-shot transfer. Accordingly, our first set of experiments consider several standard sequence-labeling tasks and datasets, namely, POS tagging the Penn Treebank (Marcus et al., 1993) with both the standard Penn Treebank POS tags and Universal POS tags (Petrov et al., 2012; Nivre et al., 2016), and the CoNLL 2003 NER task (Sang and Buchholz, 2000; Sang and De Meulder, 2003). We compare with the sequence-labeling performance of BERT (Devlin et al., 2018), which we take to be near state of the art. We use the standard dataset-splits and evaluations for all tasks, and BIO encoding for all segment-level tagging tasks.

We evaluate zero-shot transfer performance by training on one dataset and evaluating on another, without any retraining. In particular, we consider three zero-shot transfer scenarios: training with Universal POS Tags on the Penn Treebank and then predicting the standard, fine-grained POS tags, training on the CoNLL 2003 NER data and predicting on the fine-grained OntoNotes NER data (Hovy et al., 2006) using the setup of Strubell et al. (2017), and finally training on the CoNLL 2003 chunking data and predicting on the CoNLL 2003 NER data. We again compare with a BERT baseline, where labels from the original task are deterministically mapped to the most frequent label on the new task with which they coincide.[3]

Our nearest-neighbor based models were fine-tuned by retrieving the 50 nearest neighbors of each sentence in a mini-batch of either size 16 or 20, and maximizing the objective (2) above. For training, nearest neighbors were determined based on cosine-similarity between the averaged top-level (non-fine-tuned) BERT token embeddings of each sentence. In order to make training more efficient, gradients were calculated only with respect to the input sentence embeddings (i.e., the $\mathbf{x}_t$ in (1)) and not the embeddings $\mathbf{x}_k'^{(m)}$ of the tokens in $\mathcal{D}$. At test time, 100 nearest neighbors were retrieved for each sentence to be labeled using the fine-tuned embeddings.

The baseline BERT models were fine-tuned using the publicly available `huggingface` BERT implementation,[4] and the "base" weights made available by the BERT authors (Devlin et al., 2018). We made word-level predictions based on the embedding of the first tokenized word-piece associated with a word (as Devlin et al. (2018) do), and ADAM (Kingma and Ba, 2014) was used to fine-tune all models. Hyperparameters were chosen using a random search over learning rate, batch size, and number of epochs. Code for duplicating all models and experiments is available at https://github.com/swiseman/neighbor-tagging.

5 Main Results

The results of our experiments on standard sequence labeling tasks are in Table 1. We first note that all results are quite good, and are competitive with the state of the art. The label-agnostic model tends to underperform the standard fine-tuned BERT model only very slightly, though consistently, and is typically within several tenths of a point in performance.

The results of our zero-shot transfer experiments are in Table 2. We see that in all cases the label-agnostic model outperforms standard fine-

[3]For the Chunk $\rightarrow$ NER task, this results in mapping all tags to 'O', so we instead use the more favorable mapping of NPs to PERSON tags.

[4]https://github.com/huggingface/pytorch-pretrained-BERT

NER	Dev. F_1	Test F_1
BERT	95.14	90.76
NN	94.48	89.94
POS	**Dev. Acc.**	**Test Acc.**
BERT	97.56	97.91
NN	97.33	97.64
U-POS	**Dev. Acc.**	**Test Acc.**
BERT	98.34	98.62
NN	98.08	98.36

Table 1: Performance of fine-tuned BERT and nearest-neighbor based labeling (NN) on NER, POS tagging, and universal POS tagging; see text. BERT numbers are from fine-tuning the `huggingface` implementation, and differ slightly from Devlin et al. (2018).

CoNLL → Onto NER	Dev. F_1	Test F_1
BERT	58.41	58.05
NN	62.17	62.33
NN (no FT)	54.29	55.35
U-POS → POS	**Dev. Acc.**	**Test Acc.**
BERT	61.78	59.86
NN	96.70	96.98
NN (no FT)	87.44	87.13
Chunk → NER	**Dev. F_1**	**Test F_1**
BERT	9.55	8.03
NN	78.05	71.74
NN (no FT)	75.21	67.19

Table 2: Zero-shot performance of models trained on CoNLL NER and applied to fine-grained OntoNotes NER, with universal POS tags and applied to standard POS tagging, and on CoNLL chunking and applied to CoNLL NER. "NN (no FT)" indicates BERT was not fine tuned even on the original task.

tuned BERT, often significantly. In particular, we note that when going from universal POS tags to standard POS tags, the fine-tuned label-agnostic model manages to outperform the standard most-frequent-tag-per-word baseline, which itself obtains slightly less than 92% accuracy. The most dramatic increase in performance, of course, occurs on the Chunking to NER task, where the label-agnostic model is successfully able to use chunking-based training information in copying labels, whereas the parametric fine-tuned BERT model can at best attempt to map NP-chunks to PERSON labels (the most frequent named entity in the dataset).

In order to check that the increase in performance is not due only to the BERT representations themselves, Table 2 also shows the results of nearest neighbor based prediction *without* fine-tuning ("NN (no FT)" in the table) on any task. In all cases, this leads to a decrease in performance.

6 Encouraging Contiguous Copies

Although we model token-level label copying, at test time each $\hat{y}_t$ is predicted by selecting the label type with highest marginal probability, without any attempt to ensure that the resulting sequence $\hat{y}$ resembles one or a few of the labeled neighbors $y'^{(m)}$. In this section we therefore consider a decoding approach that allows for controlling the trade-off between prediction confidence and minimizing the number of *distinct* segments in $\hat{y}$ that represent direct (segment-level) copies from some neighbor, in the hope that having fewer distinct copied segments in our predictions might make them more interpretable or accurate. We emphasize that the following decoding approach is in fact applicable even to standard sequence labeling models (i.e., non-nearest-neighbor based models), as long as neighbors can be retrieved at test time.

To begin with a simple case, suppose we already know the true labels y for a sequence x, and are simply interested in being able to reconstruct y by concatenating as few segments $y'_{i:j}$ that appear in some $y'^{(m)} \in \mathcal{D}$ as possible. More precisely, define the set $\mathcal{Z}_{\mathcal{D}}$ to contain all the unique label *type* sequences appearing as a subsequence of some sequence $y'^{(m)} \in \mathcal{D}$. Then, if we're willing to tolerate some errors in reconstructing y, we can use a dynamic program to minimize the number of mislabelings in our now "prediction" $\hat{y}$, plus the number of distinct segments used in forming $\hat{y}$ multiplied by a constant c, as follows:

$$J(t) = \min_{\substack{1 \le k \le t \\ z \in \mathcal{Z}_{\mathcal{D}} : |z| = k}} J(t-k) + c + \sum_{j=1}^{k} \mathbf{1}[y_{t-k+j} \neq z_j],$$

where $J(0) = 0$ is the base case and $|z|$ is the length of sequence z. Note that greedily selecting sequences that minimize mislabelings may result in using more segments, and thus a higher J.

In the case where we do not already know y, but wish to predict it, we might consider a modification of the above, which tries to minimize c times

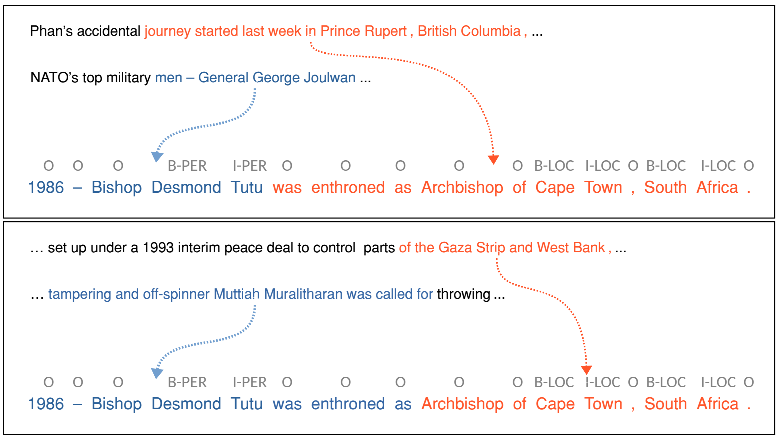

Figure 2: A CoNLL NER development example, which can be labeled with only two distinct segments. We show those used by a model trained on the NER data (top), and on chunking data and transferred zero-shot (bottom).

the number of distinct segments used in forming $\hat{y}$ plus the expected number of mislabelings:

$$J(t) = \min_{\substack{1 \le k \le t \\ z \in \mathcal{Z}_{\mathcal{D}} : |z| = k}} \Big[J(t-k) + c + \sum_{j=1}^{k} 1 - p(y_{t-k+j} = z_j \,|\, x, \mathcal{D}) \Big],$$

where we have used the linearity of expectation. Note that to use such a dynamic program to predict $\hat{y}$ we only need an estimate of $p(y_{t-k+j} = z_j \,|\, x, \mathcal{D})$, which we can obtain as in Section 3 (or from a more conventional model).

In Figure 3 we plot both the F_1 score and the average number of distinct segments used in predicting each $\hat{y}$ against the c parameter from the dynamic program above, for the CoNLL 2003 NER development data in both the standard and zero-shot settings. First we note that we are able to obtain excellent performance with only about 1.5 distinct segments per prediction, on average; see Figure 2 for examples. Interestingly, we also find that using a higher c (leading to fewer distinct segments) can in fact improve performance. Indeed, taking the best values of c from Figure 3 (0.4 in the standard setting and 0.5 in the zero-shot setting), we are able to improve our performance on the test set from 89.94 to 90.20 in the standard setting and from 71.74 to 73.61 in the zero shot setting, respectively; see Tables 1 and 2.

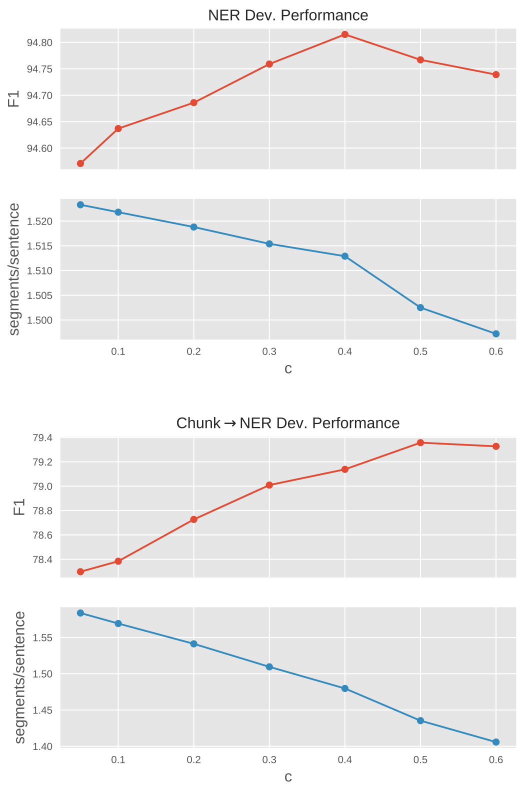

Figure 3: F_1 performance and the average number of distinct segments per predicted labeling on the CoNLL NER development data as c is varied, when training either (top) on the standard training set or (bottom) on the CoNLL chunking data (i.e., zero-shot performance).

7 Conclusion

We have proposed a simple label-agnostic sequence-labeling model, which performs nearly as well as a standard sequence labeler, but improves on zero-shot transfer tasks. We have also proposed an approach to sequence label prediction in the presence of retrieved neighbors, which allows for discouraging the use of many distinct segments in a labeling. Future work will consider problems where more challenging forms of neighbor manipulation are necessary for prediction.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853.

Claire Cardie. 1994. Domain-specific knowledge acquisition for conceptual sentence analysis. *Computer Science Department Faculty Publication Series*, page 60.

Walter Daelemans. 1993. Memory-based lexical acquisition and processing. In *Workshop on Machine Translation and Lexicon*, pages 85–98. Springer.

Walter Daelemans, Jakob Zavrel, Peter Berck, and Steven Gillis. 1996. Mbt: A memory-based part of speech tagger-generator. In *Fourth Workshop on Very Large Corpora*.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association of Computational Linguistics*, 6:437–450.

Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A retrieve-and-edit framework for predicting structured outputs. In *Advances in Neural Information Processing Systems*, pages 10073–10083.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90\% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1064–1074.

Mitchell P Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2).

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Nanyun Peng and Mark Dredze. 2016. Improving named entity recognition for chinese social media with word segmentation representation learning. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 149.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

Marek Rei and Anders Søgaard. 2018. Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. 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 293–302.

Juan Diego Rodriguez, Adam Caldwell, and Alexander Liu. 2018. Transfer learning for entity recognition of novel classes. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1974–1985.

Erik F Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*.

Tobias Schnabel and Hinrich Schütze. 2014. Flors: Fast and simple domain adaptation for part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 2:15–26.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. 2006. *Nearest-neighbor methods in learning and vision: theory and practice (neural information processing)*. The MIT press.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Jason Weston, Emily Dinan, and Alexander H Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*.

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

Ayah Zirikly and Masato Hagiwara. 2015. Cross-lingual transfer of named entity recognizers without parallel corpora. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 390–396.