

Named Entity Recognition without Labelled Data: A Weak Supervision Approach

Pierre Lison and Aliaksandr Hubin
Norwegian Computing Center
Oslo, Norway
{plison, ahu}@nr.no

Jeremy Barnes and Samia Touileb
Language Technology Group
University of Oslo, Norway
{jeremycb, samiat}@ifi.uio.no

Abstract

Named Entity Recognition (NER) performance often degrades rapidly when applied to target domains that differ from the texts observed during training. When in-domain labelled data is available, transfer learning techniques can be used to adapt existing NER models to the target domain. But what should one do when there is no hand-labelled data for the target domain? This paper presents a simple but powerful approach to learn NER models in the absence of labelled data through *weak supervision*. The approach relies on a broad spectrum of labelling functions to automatically annotate texts from the target domain. These annotations are then merged together using a hidden Markov model which captures the varying accuracies and confusions of the labelling functions. A sequence labelling model can finally be trained on the basis of this unified annotation. We evaluate the approach on two English datasets (CoNLL 2003 and news articles from Reuters and Bloomberg) and demonstrate an improvement of about 7 percentage points in entity-level F_1 scores compared to an out-of-domain neural NER model.

1 Introduction

Named Entity Recognition (NER) constitutes a core component in many NLP pipelines and is employed in a broad range of applications such as information extraction (Raiman and Raiman, 2018), question answering (Mollá et al., 2006), document de-identification (Stubbs et al., 2015), machine translation (Ugawa et al., 2018) and even conversational models (Ghazvininejad et al., 2018). Given a document, the goal of NER is to identify and classify spans referring to an entity belonging to pre-specified categories such as persons, organisations or geographical locations.

NER models often rely on convolutional or recurrent neural architectures, sometimes completed

by a CRF layer (Chiu and Nichols, 2016; Lample et al., 2016; Yadav and Bethard, 2018). More recently, deep contextualised representations relying on bidirectional LSTMS (Peters et al., 2018), transformers (Devlin et al., 2019; Yan et al., 2019) or contextual string embeddings (Akbik et al., 2019) have also been shown to achieve state-of-the-art performance on NER tasks.

These neural architectures require large corpora annotated with named entities, such as Ontonotes (Weischedel et al., 2011) or ConLL 2003 (Tjong Kim Sang and De Meulder, 2003). When only modest amounts of training data are available, transfer learning approaches can transfer the knowledge acquired from related tasks into the target domain, using techniques such as simple transfer (Rodriguez et al., 2018), discriminative fine-tuning (Howard and Ruder, 2018), adversarial transfer (Zhou et al., 2019) or layer-wise domain adaptation approaches (Yang et al., 2017; Lin and Lu, 2018).

However, in many practical settings, we wish to apply NER to domains where we have no labelled data, making such transfer learning methods difficult to apply. This paper presents an alternative approach using *weak supervision* to bootstrap named entity recognition models without requiring any labelled data from the target domain. The approach relies on labelling functions that automatically annotate documents with named-entity labels. A hidden Markov model (HMM) is then trained to unify the noisy labelling functions into a single (probabilistic) annotation, taking into account the accuracy and confusions of each labelling function. Finally, a sequence labelling model is trained using a cross-entropy loss on this unified annotation.

As in other weak supervision frameworks, the labelling functions allow us to inject *expert knowledge* into the sequence labelling model, which is often critical when data is scarce or non-existent (Hu et al., 2016; Wang and Poon, 2018). New la-

labelling functions can be easily inserted to leverage the knowledge sources at our disposal for a given textual domain. Furthermore, labelling functions can often be ported across domains, which is not the case for manual annotations that must be reiterated for every target domain.

The contributions of this paper are as follows:

1. A broad collection of labelling functions for NER, including neural models trained on various textual domains, gazetteers, heuristic functions, and document-level constraints.
2. A novel weak supervision model suited for sequence labelling tasks and able to include probabilistic labelling predictions.
3. An open-source implementation of these labelling functions and aggregation model that can scale to large datasets ¹.

2 Related Work

Unsupervised domain adaptation: Unsupervised domain adaptation attempts to adapt knowledge from a source domain to predict new instances in a target domain which often has substantially different characteristics. Earlier approaches often try to adapt the feature space using *pivots* (Blitzer et al., 2006, 2007; Ziser and Reichart, 2017) to create domain-invariant representations of predictive features. Others learn low-dimensional transformation features of the data (Guo et al., 2009; Glorot et al., 2011; Chen et al., 2012; Yu and Jiang, 2016; Barnes et al., 2018). Finally, some approaches divide the feature space into general and domain-dependent features (Daumé III, 2007). Multi-task learning can also improve cross-domain performance (Peng and Dredze, 2017).

Recently, Han and Eisenstein (2019) proposed *domain-adaptive fine-tuning*, where contextualised embeddings are first fine-tuned to both the source and target domains with a language modelling loss and subsequently fine-tuned to source domain labelled data. This approach outperforms several strong baselines trained on the target domain of the WNUT 2016 NER task (Strauss et al., 2016).

Aggregation of annotations: Approaches that aggregate annotations from multiples sources have largely concentrated on noisy data from crowd sourced annotations, with some annotators possibly

being adversarial. The *Bayesian Classifier Combination* approach of Kim and Ghahramani (2012) combines multiple independent classifiers using a linear combination of predictions. Hovy et al. (2013) learn a generative model able to aggregate crowd-sourced annotations and estimate the trustworthiness of annotators. Rodrigues et al. (2014) present an approach based on Conditional Random Fields (CRFs) whose model parameters are learned jointly using EM. Nguyen et al. (2017b) propose a Hidden Markov Model to aggregate crowd-sourced sequence annotations and find that explicitly modelling the annotator leads to improvements for POS-tagging and NER. Finally, Simpson and Gurevych (2019) proposed a fully Bayesian approach to the problem of aggregating multiple sequential annotations, using variational EM to compute posterior distributions over the model parameters.

Weak supervision: The aim of weakly supervised modelling is to reduce the need for hand-annotated data in supervised training. A particular instance of weak supervision is *distant supervision*, which relies on external resources such as knowledge bases to automatically label documents with entities that are known to belong to a particular category (Mintz et al., 2009; Ritter et al., 2013; Shang et al., 2018). Ratner et al. (2017, 2019) generalised this approach with the Snorkel framework which combines various supervision sources using a generative model to estimate the accuracy (and possible correlations) of each source. These aggregated supervision sources are then employed to train a discriminative model. Current frameworks are, however, not easily adaptable to sequence labelling tasks, as they typically require data points to be independent. One exception is the work of Wang and Poon (2018), which relies on deep probabilistic logic to perform joint inference on the full dataset. Finally, Fries et al. (2017) presented a weak supervision approach to NER in the biomedical domain. However, unlike the model proposed in this paper, their approach relies on an ad-hoc mechanism for generating candidate spans to classify.

The approach most closely related to this paper is Safranchik et al. (2020), which describe a similar weak supervision framework for sequence labelling based on an extension of HMMs called linked hidden Markov models. The authors introduce a new type of noisy rules, called linking rules, to determine how sequence elements should be grouped into spans of same tag. The main differences be-

¹<https://github.com/NorskRegnesentral/weak-supervision-for-NER>.

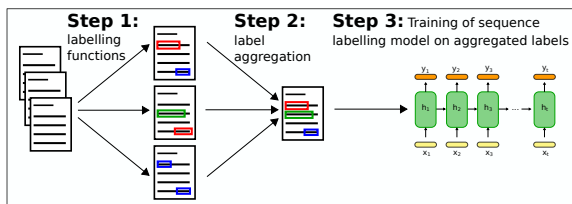


Figure 1: Illustration of the weak supervision approach.

tween their approach and this paper are the linking rules, which are not employed here, and the choice of labelling functions, in particular the document-level relations detailed in Section 3.1.

Ensemble learning: The proposed approach is also loosely related to *ensemble methods* such as bagging, boosting and random forests (Sagi and Rokach, 2018). These methods rely on multiple classifiers run simultaneously and whose outputs are combined at prediction time. In contrast, our approach (as in other weak supervision frameworks) only requires labelling functions to be aggregated once, as an intermediary step to create training data for the final model. This is a non-trivial difference as running all labelling functions at prediction time is computationally costly due to the need to run multiple neural models along with gazetteers extracted from large knowledge bases.

3 Approach

The proposed model collects weak supervision from multiple *labelling functions*. Each labelling function takes a text document as input and outputs a series of spans associated with NER labels. These outputs are then aggregated using a hidden Markov model (HMM) with multiple emissions (one per labelling function) whose parameters are estimated in an unsupervised manner. Finally, the aggregated labels are employed to learn a sequence labelling model. Figure 1 illustrates this process. The process is performed on documents from the target domain, e.g. a corpus of financial news.

Labelling functions are typically *specialised* to detect only a subset of possible labels. For instance, a gazetteer based on Wikipedia will only detect mentions of persons, organisations and geographical locations and ignore entities such as dates or percents. This marks a departure from existing aggregation methods, which are originally designed for crowd-sourced data and where annotators are supposed to make use of the full label set. In addition, unlike previous weak supervision approaches,

we allow labelling functions to produce *probabilistic predictions* instead of deterministic values. The aggregation model described in Section 3.2 directly captures these properties in the emission model associated with each labelling function.

We first briefly describe the labelling functions integrated into the current system. We review in Section 3.2 the aggregation model employed to combine the labelling predictions. The final labelling model is presented in Section 3.3. The complete list of 52 labelling functions employed in the experiments is available in Appendix A.

3.1 Labelling functions

Out-of-domain NER models The first set of labelling functions are sequence labelling models trained in domains from which labelled data is available. In the experiments detailed in Section 4, we use four such models, respectively trained on Ontonotes (Weischedel et al., 2011), CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003)², the Broad Twitter Corpus (Derczynski et al., 2016) and a NER-annotated corpus of SEC filings (Salinas Alvarado et al., 2015).

For the experiments in this paper, all aforementioned models rely on a transition-based NER model (Lample et al., 2016) which extracts features with a stack of four convolutional layers with filter size of three and residual connections. The model uses attention features and a multi-layer perceptron to select the next transition. It is initialised with GloVe embeddings (Pennington et al., 2014) and implemented in Spacy (Honnibal and Montani, 2017). However, the proposed approach does not impose any constraints on the model architecture and alternative approaches based on e.g. contextualised embeddings can also be employed.

Gazetteers As in distant supervision approaches, we include a number of gazetteers from large knowledge bases to identify named entities. Concretely, we use resources from Wikipedia (Geiß et al., 2018), Geonames (Wick, 2015), the Crunchbase Open Data Map, DBpedia (Lehmann et al., 2015) along with lists of countries, languages, nationalities and religious or political groups.

To efficiently search for occurrences of these entities in large text collections, we first convert each knowledge base into a *trie* data structure. Prefix search is then applied to extract matches (using

²The CoNLL 2003 NER model is of course deactivated for the experimental evaluation on CoNLL 2003.

both case-sensitive and case-insensitive mode, as they have distinct precision-recall trade-offs).

Heuristic functions We also include various heuristic functions, each specialised in the recognition of specific types of named entities. Several functions are dedicated to the recognition of proper names based on casing, part-of-speech tags or dependency relations. In addition, we integrate a variety of handcrafted functions relying on regular expressions to detect occurrences of various entities (see Appendix A for details). A probabilistic parser specialised in the recognition of dates, times, money amounts, percents, and cardinal/ordinal values (Braun et al., 2017) is also incorporated.

Document-level relations All labelling functions described above rely on local decisions on tokens or phrases. However, texts are not loose collections of words, but exhibit a high degree of internal coherence (Grosz and Sidner, 1986; Grosz et al., 1995) which can be exploited to further improve the annotations.

We introduce one labelling function to capture label consistency constraints in a document. As noted in (Krishnan and Manning, 2006; Wang et al., 2018), named entities occurring multiple times through a document have a high probability of belonging to the same category. For instance, while *Komatsu* may both refer to a Japanese town or a multinational corporation, a text including this mention will either be about the town or the company, but rarely both at the same time. To capture these non-local dependencies, we define the following label consistency model: given a text span e occurring in a given document, we look for all spans Z_e in the document that contain the same string as e . The (probabilistic) output of the labelling function then corresponds to the relative frequency of each label l for that string in the document:

$$P_{\text{doc_majority}(e)}(l) = \frac{\sum_{z \in Z_e} P_{\text{label}(z)}(l)}{|Z_e|} \quad (1)$$

The above formula depends on a distribution $P_{\text{label}(z)}$, which can be defined on the basis of other labelling functions. Alternatively, a two-stage model similar to (Krishnan and Manning, 2006) could be employed to first aggregate local labelling functions and subsequently apply document-level functions on aggregated predictions.

Another insight from Grosz and Sidner (1986) is the importance of the *attentional structure*. When

introduced for the first time, named entities are often referred to in an explicit and univocal manner, while subsequent mentions (once the entity is a part of the focus structure) frequently rely on shorter references. The first mention of a person in a given text is for instance likely to include the person’s full name, and is often shortened to the person’s last name in subsequent mentions. As in Ratnov and Roth (2009), we determine whether a proper name is a substring of another entity mentioned earlier in the text. If so, the labelling function replicates the label distribution of the first entity.

3.2 Aggregation model

The outputs of these labelling functions are then aggregated into a single layer of annotation through an *aggregation model*. As we do not have access to labelled data for the target domain, this model is estimated in a fully unsupervised manner.

Model We assume a list of J labelling functions $\{\lambda_1, \dots, \lambda_J\}$ and a list of S mutually exclusive NER labels $\{l_1, \dots, l_S\}$. The aggregation model is represented as an HMM, in which the states correspond to the true underlying labels. This model has multiple emissions (one per labelling function) assumed to be mutually independent conditional on the latent underlying label.

Formally, for each token $i \in \{1, \dots, n\}$ and labelling function j , we assume a Dirichlet distribution for the probability labels P_{ij} . The parameters of this Dirichlet are separate vectors $\alpha_j^{s_i} \in \mathcal{R}_{[0,1]}^S$, for each of the latent states $s_i \in \{1, \dots, S\}$. The latent states are assumed to have a Markovian dependence structure between the tokens $\{1, \dots, n\}$. This results in the HMM represented by a dependent mixtures of Dirichlet model:

$$P_{ij} | \alpha_j^{s_i} \stackrel{\text{ind}}{\sim} \text{Dirichlet}(\alpha_j^{s_i}), \quad (2)$$

$$p(s_i | s_{i-1}) = \text{logit}^{-1}(\omega^{(s_i, s_{i-1})}), \quad (3)$$

$$\text{logit}^{-1}(\omega^{(s_i, s_{i-1})}) = \frac{e^{\omega^{(s_i, s_{i-1})}}}{1 + e^{\omega^{(s_i, s_{i-1})}}}. \quad (4)$$

Here, $\omega^{(s_i, s_{i-1})} \in \mathcal{R}$ are the parameters of the transition probability matrix controlling for a given state s_{i-1} the probability of transition to state s_i . Figure 2 illustrates the model structure.

Parameter estimation The learnable parameters of this HMM are (a) the transition matrix between states and (b) the α vectors of the Dirichlet distribution associated with each labelling function. The

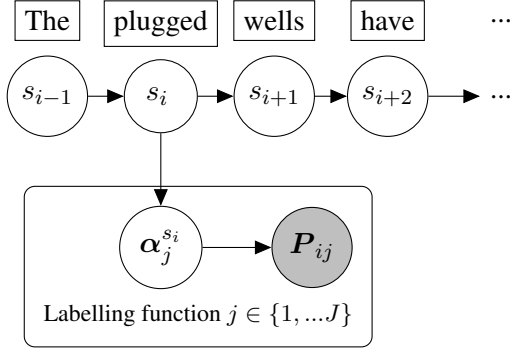


Figure 2: Aggregation model using a hidden Markov model with multiple probabilistic emissions.

transition matrix is of size $|S| \times |S|$, while we have $|S| \times |J|$ α vectors, each of size $|S|$. The parameters are estimated with the Baum-Welch algorithm, which is a variant of EM algorithm that relies on the forward-backward algorithm to compute the statistics for the expectation step.

To ensure faster convergence, we introduce a new constraint to the likelihood function: for each token position i , the corresponding latent label s_i must have a non-zero probability in at least one labelling function (the likelihood of this label is otherwise set to zero for that position). In other words, the aggregation model will only predict a particular label if this label is produced by least one labelling function. This simple constraint facilitates EM convergence as it restricts the state space to a few possible labels at every time-step.

Prior distributions The HMM described above can be provided with informative priors. In particular, the initial distribution for the latent states can be defined as a Dirichlet based on counts δ for the most reliable labelling function³:

$$p(s_i) \stackrel{d}{=} \text{Dirichlet}(\delta). \quad (5)$$

The prior for each row k of the transition probabilities matrix is also a Dirichlet based on the frequencies of transitions between the observed classes for the most reliable labelling function κ_k :

$$p(s_i | s_{i-1} = k) \stackrel{d}{=} \text{Dirichlet}(\kappa_k). \quad (6)$$

Finally, to facilitate convergence of the EM algorithm, informative starting values can be specified for the emission model of each labelling function.

³The most reliable labelling function was found in our experiments to be the NER model trained on Ontonotes 5.0.

Assuming we can provide rough estimates of the recall r_{jk} and precision ρ_{jk} for the labelling function j on label k , the initial values for the parameters of the emission model are expressed as:

$$\alpha_{jk}^{s_i} \propto \begin{cases} r_{jk}, & \text{if } s_i = k, \\ (1 - r_{s_ik}) (1 - \rho_{jk}) \delta_k, & \text{if } s_i \neq k. \end{cases}$$

The probability of observing a given label k emitted by the labelling function j is thus proportional to its recall if the true label is indeed k . Otherwise (i.e. if the labelling function made an error), the probability of emitting k is inversely proportional to the precision of the labelling function j .

Decoding Once the parameters of the HMM model are estimated, the forward-backward algorithm can be employed to associate each token marginally with a posterior probability distribution over possible NER labels (Rabiner, 1990).

3.3 Sequence labelling model

Once the labelling functions are aggregated on documents from the target domain, we can train a sequence labelling model on the unified annotations, without imposing any constraints on the type of model to use. To take advantage of the posterior marginal distribution \tilde{p}_s over the latent labels, the optimisation should seek to minimise the expected loss with respect to \tilde{p}_s :

$$\hat{\theta} = \arg \min_{\theta} \sum_i^n \mathbb{E}_{y \sim \tilde{p}_s} [\text{loss}(h_{\theta}(x_i), y)] \quad (7)$$

where $h_{\theta}(\cdot)$ is the output of the sequence labelling model. This is equivalent to minimising the cross-entropy error between the outputs of the neural model and the probabilistic labels produced by the aggregation model.

4 Evaluation

We evaluate the proposed approach on two English-language datasets, namely the CoNLL 2003 dataset and a collection of sentences from Reuters and Bloomberg news articles annotated with named entities by crowd-sourcing. We include a second dataset in order to evaluate the approach with a more fine-grained set of NER labels than the ones in CoNLL 2003. As the objective of this paper is to compare approaches to unsupervised domain adaptation, we do not rely on any labelled data from these two target domains.

4.1 Data

CoNLL 2003 The CoNLL 2003 dataset (Tjong Kim Sang and De Meulder, 2003) consists of 1163 documents, including a total of 35089 entities spread over 4 labels: ORG, PER, LOC and MISC.

Reuters & Bloomberg We additionally crowd annotate 1054 sentences from Reuters and Bloomberg news articles from Ding et al. (2014). We instructed the annotators to tag sentences with the following 9 Ontonotes-inspired labels: PERSON, NORP, ORG, LOC, PRODUCT, DATETIME, PERCENT, MONEY, QUANTITY. Note that the DATE and TIME labels from Ontonotes are merged into DATETIME, and the LOC and GPE labels are similarly merged into LOC. Each sentence was annotated by at least two annotators, and a qualifying test with gold-annotated questions was conducted for quality control. Cohen’s κ for sentences with two annotators is 0.39, while Krippendorff’s α for three annotators is 0.44. We had to remove QUANTITY labels from the annotations as the crowd results for this particular label were highly inconsistent.

4.2 Baselines

Ontonotes-trained NER The first baseline corresponds to a neural sequence labelling model trained on the Ontonotes 5.0 corpus. We use here the same model from Section 3.1, which is the single best-performing labelling function (that is, without aggregating multiple predictions).

We also experimented with other neural architectures but these performed similar or worse than the transition-based model, presumably because they are more prone to overfitting on the source domain.

Majority voting (MV) The simplest method for aggregating outputs is majority voting, i.e. outputting the most frequent label among the ones predicted by each labelling function. However, specialised labelling functions will output O for most tokens, which means that the majority label is typically O. To mitigate this problem, we first look at tokens that are marked with a non-O label by at least T labelling functions (where T is a hyperparameter tuned experimentally), and then apply majority voting on this set of non-O labels.

Snorkel model The Snorkel framework (Ratner et al., 2017) does not directly support sequence labelling tasks as data points are required to be independent. However, heuristics can be used to

extract named-entity candidates and then apply labelling functions to infer their most likely labels (Fries et al., 2017). For this baseline, we use the three functions `nnp_detector`, `proper_detector` and `compound_detector` (see Appendix A) to generate candidate spans. We then create a matrix expressing the prediction of each labelling function for each span (including a specific “abstain” value to denote the absence of predictions) and run the matrix-completion-style approach of Ratner et al. (2019) to aggregate the predictions.

mSDA is a strong domain adaptation baseline (Chen et al., 2012) which augments the feature space of a model with intermediate representations learned using stacked denoising autoencoders. In our case, we learn the mSDA representations on the unlabeled source and target domain data. These 800 dimensional vectors are concatenated to 300 dimensional word embeddings and fed as input to a two-layer LSTM with a skip connection. Finally, we train the LSTM on the labeled source data and test on the target domain.

AdaptaBERT This baseline corresponds to a state-of-the-art unsupervised domain adaptation approach (AdaptaBERT) (Han and Eisenstein, 2019). The approach first uses unlabeled data from both the source and target domains to domain-tune a pretrained BERT model. The model is finally task-tuned in a supervised fashion on the source domain labelled data (Ontonotes). At inference time, the model is able to make use of the pretraining and domain tuning to predict entities in the target domain. In our experiments, we use the cased-version of the base BERT model (trained on Wikipedia and news text) and perform three fine-tuning epochs for both domain-tuning and task-tuning. We additionally include an ensemble model, which averages the predictions of five BERT models fine-tuned with different random seeds.

Mixtures of multinomials

Following the notation from Section 3.2, we define $Y_{i,j,k} = \mathbb{I}(P_{i,j,k} = \max_{k' \in \{1, \dots, S\}} P_{i,j,k'})$ to be the most probable label for word i by source j . One can model $Y_{i,j}$ with a Multinomial probability distribution. The first four baselines (the fifth one assumes Markovian dependence between the latent states) listed below use the following independent, i.e. $p(s_i, s_{i-1}) = p(s_i)p(s_{i-1})$, mixtures

of Multinomials model for Y_{ij} :

$$\begin{aligned} Y_{ij}|p_j^{s_i} &\overset{ind}{\sim} \text{Multinomial}(p_j^{s_i}), \\ s_i &\overset{ind}{\sim} \text{Multinomial}(\sigma). \end{aligned}$$

Accuracy model (ACC) (Rodrigues et al., 2014) assumes the following constraints on $p_j^{s_i}$:

$$p_{jk}^{s_i} = \begin{cases} \pi_j, & \text{if } s_i = k, \\ \frac{1-\pi_j}{J-1} s_i \neq k. \end{cases}$$

Here, for each labelling function it is assumed to have the same accuracy π_j for all of the tokens.

Confusion vector (CV) (Nguyen et al., 2017a) extends ACC by relying on separate success probabilities for each token label:

$$p_{jk}^{s_i} = \begin{cases} \pi_{jk}, & \text{if } s_i = k, \\ \frac{1-\pi_{jk}}{J-1} s_i \neq k. \end{cases}$$

Confusion matrix (CM) (Dawid and Skene, 1979) allows for distinct accuracies conditional on the latent states, which results in:

$$p_{jk}^{s_i} = \pi_{jk}^{s_i}. \quad (8)$$

Sequential Confusion Matrix (SEQ) extends the CM model of Simpson and Gurevych (2019), where an "auto-regressive" component is included in the observed part of the model. We assume dependence on a covariate indicating that the label has not changed for a given source, i.e.:

$$p_{jk}^{s_i} = \text{logit}^{-1}(\mu_{jk}^{s_i} + \mathbf{I}(Y_{i-1,j,k}^T = Y_{i,j,k}^T)\beta_{jk}^{s_i}).$$

Dependent confusion matrix (DCM) combines the CM-distinct accuracies conditional on the latent states of (8) and the Markovian dependence of (3).

4.3 Results

The evaluation results are shown in Tables 1 and 2, respectively for the CoNLL 2003 data and the sentences extracted from Reuters and Bloomberg. The metrics are the (micro-averaged) precision, recall and F_1 scores at both the token-level and entity-level. In addition, we indicate the token-level cross-entropy error (in log-scale). As the labelling functions are defined on a richer annotation scheme than the four labels of CoNLL 2003, we map GPE to LOC and EVENT, FAC, LANGUAGE, LAW, NORP, PRODUCT and WORK_OF_ART to MISC.

The results for the ACC and CV baselines are not included in the two tables as the parameter estimation did not converge (and thus did not provide reliable estimates of the parameters).

Table 1 further details the results obtained using only a subset of labelling functions. Of particular interest is the positive contribution of document-level functions, boosting the entity-level F_1 from 0.702 to 0.716. This highlights the importance of document-level relations in NER.

The last line of the two tables reports the performance of the neural sequence labelling model (described in Section 3.3) trained on the basis of the aggregated labels. We observe that the performance of this neural model remains close to the performance of the HMM-aggregated labels. This result shows that the knowledge from the labelling functions can be injected into a standard neural model without substantial loss.

4.4 Discussion

Although not shown in the results due to space constraints, we also analysed whether the informative priors described in Section 3.2 influenced the performance of the aggregation model. We found informative and non-informative priors to yield similar performance for CoNLL 2003. However, the performance of non-informative priors was very poor on the Reuters and Bloomberg sentences (F_1 at 0.12), thereby demonstrating the usefulness of informative priors for small datasets.

We provide in Figure 3 an example with a few selected labelling functions. In particular, we can observe that the Ontonotes-trained NER model mistakenly labels "Heidrun" as a product. This erroneous label, however, is counter-balanced by other labelling functions, notably a document-level function looking at the global label frequency of this string through the document. We do, however, notice a few remaining errors, e.g. the labelling of "Status Weekly" as an organisation.

Figure 4 illustrates the pairwise agreement and disagreement between labelling functions on the CoNLL 2003 dataset. If both labelling functions make the same prediction on a given token, we count this as an agreement, whereas conflicting predictions (ignoring O labels), are seen as disagreement. Large differences may exist between these functions for specific labels, especially MISC. The functions with the highest overlap are those making predictions on all labels, while labelling functions

Model:	Token-level				Entity-level		
	P	R	F_1	CEE	P	R	F_1
Ontonotes-trained NER	0.719	0.706	0.712	2.671	0.694	0.620	0.654
Majority voting (MV)	0.815	0.675	0.738	2.047	0.751	0.619	0.678
Confusion Matrix (CM)	0.786	0.746	0.766	1.964	0.713	0.700	0.706
Sequential Confusion Matrix (SEQ)	0.736	0.716	0.726	2.254	0.642	0.668	0.654
Dependent Confusion Matrix (DCM)	0.785	0.744	0.764	1.983	0.710	0.698	0.704
Snorkel-aggregated labels	0.710	0.661	0.684	2.264	0.714	0.621	0.664
mSDA (OntoNotes)	0.640	0.569	0.603	2.813	0.560	0.562	0.561
AdaptaBERT (OntoNotes)	0.693	0.733	0.712	2.280	0.652	0.736	0.691
AdaptaBERT (Ensemble)	0.704	0.754	0.729	2.103	0.684	0.743	0.712
HMM-aggregated labels (only NER models)	0.658	0.720	0.688	2.653	0.642	0.599	0.620
HMM-aggregated labels (only gazetteers)	0.759	0.394	0.518	3.678	0.687	0.367	0.478
HMM-aggregated labels (only heuristics)	0.722	0.771	0.746	1.989	0.718	0.683	0.700
HMM-aggregated labels (all but doc-level)	0.714	0.778	0.744	1.878	0.713	0.693	0.702
HMM-aggregated labels (all functions)	0.719	0.794	0.754	1.812	0.721	0.713	0.716
Neural net trained on HMM-agg. labels	0.712	0.790	0.748	2.282	0.715	0.707	0.710

Table 1: Evaluation results on CoNLL 2003. MV=Majority Voting, P=Precision, R=Recall, CEE=Cross-entropy Error (lower is better). The results are micro-averaged on all labels (PER, ORG, LOC and MISC).

Model:	Token-level				Entity-level		
	P	R	F_1	CEE	P	R	F_1
OntoNotes-trained NER	0.793	0.791	0.792	2.648	0.694	0.635	0.664
Majority voting (MV)	0.832	0.713	0.768	2.454	0.699	0.644	0.670
Confusion Matrix (CM)	0.816	0.702	0.754	2.708	0.667	0.636	0.652
Sequential Confusion Matrix (SEQ)	0.741	0.630	0.682	3.261	0.535	0.547	0.540
Dependent Confusion Matrix (DCM)	0.819	0.706	0.758	2.702	0.673	0.641	0.656
mSDA (OntoNotes)	0.749	0.751	0.750	2.501	0.618	0.684	0.649
AdaptaBERT (OntoNotes)	0.799	0.801	0.800	2.351	0.668	0.734	0.699
AdaptaBERT (Ensemble)	0.813	0.815	0.814	2.265	0.682	0.748	0.713
HMM-aggregated labels (all functions)	0.804	0.823	0.814	2.219	0.749	0.697	0.722
Neural net trained on HMM-agg. labels	0.805	0.827	0.816	2.448	0.749	0.701	0.724

Table 2: Evaluation results on 1094 crowd-annotated sentences from Reuters and Bloomberg news articles. The results are micro-averaged on 8 labels (PERSON, NORP, ORG, LOC, PRODUCT, DATE, PERCENT, and MONEY).

specialised to few labels (such as legal.detector) often have less overlap. We also observe that the two gazetteers from Crunchbase and Geonames disagree in about 15% of cases, presumably due to company names that are also geographical locations, as in the earlier Komatsu example.

In terms of computational efficiency, the estimation of HMM parameters is relatively fast, requiring less than 30 mins on the entire CoNLL 2003 data. Once the aggregation model is estimated, it

can be directly applied to new texts with a single forward-backward pass, and can therefore scale to datasets with hundreds of thousands of documents. This runtime performance is an important advantage compared to approaches such as AdaptaBERT (Han and Eisenstein, 2019) which are relatively slow at inference time. The proposed approach can also be ported to other languages than English, although heuristic functions and gazetteers will need to be adapted to the target language.

Well repairs to lift Heidrun oil output - Statoil . OSLO 1996-08-22 Three plugged water injection wells on the Heidrun oilfield off mid-Norway will be reopened over the next month , operator Den Norske Stats Oljeselskap AS (Statoil) said on Thursday .

The plugged wells have accounted for a dip of 30,000 barrels per day (bpd) in Heidrun output to roughly 220,000 bpd , according to the company 's Status Weekly newsletter . The wells will be reperforated and gravel will be pumped into the reservoir through one of the wells to avoid plugging problems in the future , it said . - Oslo newsroom

Neural models: Ontonotes-trained NER ; Gazetteers: company_uncased ; Heuristic functions: date_detector, snips, and number_detector ; Document level functions: doc_majority_uncased ; Aggregated predictions: HMM-aggregated model

Figure 3: Extended example showing the outputs of 6 labelling functions, along with the HMM-aggregated model.

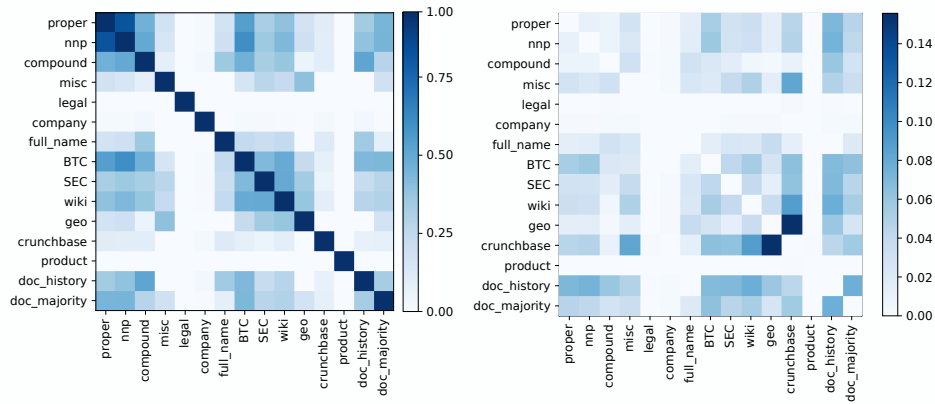


Figure 4: Pairwise agreement (left) and disagreement (right) between the labelling functions on the CoNLL 2003 data with labels PER, ORG, LOC, MISC, normalized by total number of labelled examples.

5 Conclusion

This paper presented a weak supervision model for sequence labelling tasks such as Named Entity Recognition. To leverage all possible knowledge sources available for the task, the approach uses a broad spectrum of labelling functions, including data-driven NER models, gazetteers, heuristic functions, and document-level relations between entities. Labelling functions may be specialised to recognise specific labels while ignoring others. Furthermore, unlike previous weak supervision approaches, labelling functions may produce probabilistic predictions. The outputs of these labelling functions are then merged together using a hidden Markov model whose parameters are estimated with the Baum-Welch algorithm. A neural sequence labelling model can finally be learned on the basis of these unified predictions.

Evaluation results on two datasets (CoNLL 2003 and news articles from Reuters and Bloomberg)

show that the method can boost NER performance by about 7 percentage points on entity-level F_1 . In particular, the proposed model outperforms the unsupervised domain adaptation approach through contextualised embeddings of [Han and Eisenstein \(2019\)](#). Of specific linguistic interest is the contribution of document-level labelling functions, which take advantage of the internal coherence and narrative structure of the texts.

Future work will investigate how to take into account potential correlations between labelling functions in the aggregation model, as done in e.g. [\(Bach et al., 2017\)](#). Furthermore, some of the labelling functions can be rather noisy and model selection of the optimal subset of the labelling functions might well improve the performance of our model. Model selection approaches that can be adapted are discussed in [Adams and Beling \(2019\)](#); [Hubin \(2019\)](#). We also wish to evaluate the approach on other types of sequence labelling tasks beyond Named Entity Recognition.

Acknowledgements

The research presented in this paper was conducted as part of the innovation project "FinAI: Artificial Intelligence tool to monitor global financial markets" in collaboration with Exabel AS⁴. This collaboration is supported through the funding programme for "User-driven Research based Innovation" of the Research Council of Norway.

Additionally, this work is supported by the SANT project (Sentiment Analysis for Norwegian Text), funded by the Research Council of Norway (grant number 270908)

References

- Stephen Adams and Peter A Beling. 2019. A survey of feature selection methods for Gaussian mixture models and hidden Markov models. *Artificial Intelligence Review*, 52(3):1739–1779.
- Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. [Pooled contextualized embeddings for named entity recognition](#). 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 724–728, Minneapolis, Minnesota. Association for Computational Linguistics.
- Stephen H. Bach, Bryan He, Alexander Ratner, and Christopher Ré. 2017. [Learning the structure of generative models without labeled data](#). In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, pages 273–282. JMLR.org.
- Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. 2018. [Projecting embeddings for domain adaptation: Joint modeling of sentiment analysis in diverse domains](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 818–830, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- John Blitzer, Mark Dredze, and Fernando Pereira. 2007. [Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification](#). In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.
- John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. [Domain adaptation with structural correspondence learning](#). In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia. Association for Computational Linguistics.
- Daniel Braun, Adrian Hernandez Mendez, Florian Matthes, and Manfred Langen. 2017. [Evaluating natural language understanding services for conversational question answering systems](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 174–185, Saarbrücken, Germany. Association for Computational Linguistics.
- Minmin Chen, Zhixiang Xu, Kilian Q. Weinberger, and Fei Sha. 2012. [Marginalized denoising autoencoders for domain adaptation](#). In *Proceedings of the 29th International Conference on International Conference on Machine Learning, ICML'12*, pages 1627–1634, USA. Omnipress.
- Jason P.C. Chiu and Eric Nichols. 2016. [Named entity recognition with bidirectional LSTM-CNNs](#). *Transactions of the Association for Computational Linguistics*, 4:357–370.
- Hal Daumé III. 2007. [Frustratingly easy domain adaptation](#). In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.
- A. P. Dawid and A. M. Skene. 1979. [Maximum likelihood estimation of observer error-rates using the em algorithm](#). *Applied Statistics*, 28(1):20–28.
- Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. [Broad twitter corpus: A diverse named entity recognition resource](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan. The COLING 2016 Organizing Committee.
- 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.
- Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2014. [Using structured events to predict stock price movement: An empirical investigation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1415–1425, Doha, Qatar. Association for Computational Linguistics.
- Jason Fries, Sen Wu, Alex Ratner, and Christopher Ré. 2017. [Swellshark: A generative model for biomedical named entity recognition without labeled data](#).
- Johanna Geiß, Andreas Spitz, and Michael Gertz. 2018. [Neckar: A named entity classifier for wikidata](#). In *Language Technologies for the Challenges of the Digital Age*, pages 115–129, Cham. Springer International Publishing.

⁴www.exabel.com

- Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Scott Wen-tau Yih, and Michel Galley. 2018. [A knowledge-grounded neural conversation model](#). In *AAAI*.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. [Domain adaptation for large-scale sentiment classification: A deep learning approach](#). In *Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11*, pages 513–520, USA. Omnipress.
- Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. [Centering: A framework for modeling the local coherence of discourse](#). *Computational Linguistics*, 21(2):203–225.
- Barbara J. Grosz and Candace L. Sidner. 1986. [Attention, intentions, and the structure of discourse](#). *Computational Linguistics*, 12(3):175–204.
- Honglei Guo, Huijia Zhu, Zhili Guo, Xiaoxun Zhang, Xian Wu, and Zhong Su. 2009. [Domain adaptation with latent semantic association for named entity recognition](#). In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 281–289, Boulder, Colorado. Association for Computational Linguistics.
- Xiaochuang Han and Jacob Eisenstein. 2019. [Unsupervised domain adaptation of contextualized embeddings for sequence labeling](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4237–4247, Hong Kong, China. Association for Computational Linguistics.
- Matthew Honnibal and Ines Montani. 2017. *spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing*. To appear.
- Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. [Learning whom to trust with MACE](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.
- Jeremy Howard and Sebastian Ruder. 2018. [Universal language model fine-tuning for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.
- Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. [Harnessing deep neural networks with logic rules](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.
- Aliaksandr Hubin. 2019. An adaptive simulated annealing EM algorithm for inference on non-homogeneous hidden Markov models. In *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, pages 1–9.
- Hyun-Chul Kim and Zoubin Ghahramani. 2012. [Bayesian classifier combination](#). In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 619–627, La Palma, Canary Islands. PMLR.
- Vijay Krishnan and Christopher D. Manning. 2006. [An effective two-stage model for exploiting non-local dependencies in named entity recognition](#). In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1121–1128, Sydney, Australia. 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.
- Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. [Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia](#). *Semantic Web*, 6(2):167–195.
- Bill Yuchen Lin and Wei Lu. 2018. [Neural adaptation layers for cross-domain named entity recognition](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2012–2022, Brussels, Belgium. Association for Computational Linguistics.
- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. [Distant supervision for relation extraction without labeled data](#). In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.
- Diego Mollá, Menno van Zaanen, and Daniel Smith. 2006. [Named entity recognition for question answering](#). In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58, Sydney, Australia.
- An T Nguyen, Byron C Wallace, Junyi Jessy Li, Ani Nenkova, and Matthew Lease. 2017a. Aggregating and predicting sequence labels from crowd annotations. In *Proceedings of the conference. Asso-*

- ciation for Computational Linguistics. Meeting, volume 2017, page 299. NIH Public Access.
- An Thanh Nguyen, Byron Wallace, Junyi Jessy Li, Ani Nenkova, and Matthew Lease. 2017b. [Aggregating and predicting sequence labels from crowd annotations](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 299–309, Vancouver, Canada. Association for Computational Linguistics.
- Nanyun Peng and Mark Dredze. 2017. [Multi-task domain adaptation for sequence tagging](#). In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 91–100, Vancouver, Canada. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- 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)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Lawrence R. Rabiner. 1990. In Alex Waibel and Kai-Fu Lee, editors, *Readings in Speech Recognition*, chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. [\[link\]](#).
- Jonathan Raiman and Olivier Raiman. 2018. [Deep-type: Multilingual entity linking by neural type system evolution](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5406–5413.
- Lev Ratinov and Dan Roth. 2009. [Design challenges and misconceptions in named entity recognition](#). In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.
- Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. [Snorkel: Rapid training data creation with weak supervision](#). *Proc. VLDB Endow.*, 11(3):269–282.
- Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2019. [Snorkel: rapid training data creation with weak supervision](#). *The VLDB Journal*.
- Alan Ritter, Luke Zettlemoyer, Mausam, and Oren Etzioni. 2013. [Modeling missing data in distant supervision for information extraction](#). *Transactions of the Association for Computational Linguistics*, 1:367–378.
- Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. [Sequence labeling with multiple annotators](#). *Mach. Learn.*, 95(2):165–181.
- 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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Esteban Safranchik, Shiyong Luo, and Stephen H. Bach. 2020. [Weakly supervised sequence tagging from noisy rules](#). In *AAAI Conference on Artificial Intelligence (AAAI)*.
- Omer Sagi and Lior Rokach. 2018. [Ensemble learning: A survey](#). *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249.
- Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. 2015. [Domain adaption of named entity recognition to support credit risk assessment](#). In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90, Parramatta, Australia.
- Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. [Learning named entity tagger using domain-specific dictionary](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064, Brussels, Belgium. Association for Computational Linguistics.
- Edwin D. Simpson and Iryna Gurevych. 2019. [A Bayesian approach for sequence tagging with crowds](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1093–1104, Hong Kong, China. Association for Computational Linguistics.
- Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. [Results of the WNUT16 named entity recognition shared task](#). In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan. The COLING 2016 Organizing Committee.
- Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. 2015. [Automated systems for the de-identification of longitudinal clinical narratives](#). *Journal of Biomedical Informatics*, 58(S):S11–S19.

- 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*, pages 142–147.
- Arata Ugawa, Akihiro Tamura, Takashi Ninomiya, Hiroya Takamura, and Manabu Okumura. 2018. [Neural machine translation incorporating named entity](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3240–3250, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Hai Wang and Hoifung Poon. 2018. [Deep probabilistic logic: A unifying framework for indirect supervision](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1891–1902, Brussels, Belgium. Association for Computational Linguistics.
- Limin Wang, Shoushan Li, Qian Yan, and Guodong Zhou. 2018. [Domain-specific named entity recognition with document-level optimization](#). *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 17(4):33:1–33:15.
- R. Weischedel, E. Hovy, M. Marcus, Palmer M., R. Belvin, S. Pradhan, L. Ramshaw, and N. Xue. 2011. OntoNotes: A large training corpus for enhanced processing. In *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. Springer.
- Marc Wick. 2015. [Geonames ontology](#).
- Vikas Yadav and Steven Bethard. 2018. [A survey on recent advances in named entity recognition from deep learning models](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. Tener: Adapting transformer encoder for name entity recognition. *arXiv preprint arXiv:1911.04474*.
- Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. [Transfer learning for sequence tagging with hierarchical recurrent networks](#). In *International Conference on Learning Representations*.
- Jianfei Yu and Jing Jiang. 2016. [Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 236–246, Austin, Texas. Association for Computational Linguistics.
- Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. 2019. [Dual adversarial neural transfer for low-resource named entity recognition](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3461–3471, Florence, Italy. Association for Computational Linguistics.
- Yftah Ziser and Roi Reichart. 2017. [Neural structural correspondence learning for domain adaptation](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 400–410, Vancouver, Canada. Association for Computational Linguistics.

A Labelling functions

Group	Function name	Description
Neural NER models	BTC	Model trained on the Broad Twitter Corpus
	BTC+c	Model trained on the Broad Twitter Corpus + postprocessing
	SEC	Model trained on SEC-filings
	SEC+c	Model trained on SEC-filings + postprocessing
	conll2003	Model trained on CoNLL 2003
	conll2003+c	Model trained on CoNLL 2003 + postprocessing
	core_web_md	Model trained on Ontonotes 5.0
	core_web_md+c	Model trained on Ontonotes 5.0 + postprocessing
Gazetteers	wiki_cased	Gazetteer (case-sensitive) using Wikipedia entries
	multitoken_wiki_cased	Same as above, but restricted to multitoken entities
	wiki_uncased	Gazetteer (case-insensitive) using Wikipedia entries
	multitoken_wiki_uncased	Same as above, but restricted to multitoken entities
	wiki_small_cased	Gazetteer (case-sensitive) using Wikipedia entries with non-empty description
	multitoken_wiki_small_cased	Same as above, but restricted to multitoken entities
	wiki_small_uncased	Gazetteer (case-insensitive) using Wikipedia entries with non-empty description
	multitoken_wiki_small_uncased	Same as above, but restricted to multitoken entities
	company_cased	Gazetteer (case-sensitive) using a large list of company names
	multitoken_company_cased	Same as above, but restricted to multitoken entities
	company_uncased	Gazetteer from a large list of company names (case-insensitive)
	multitoken_company_uncased	Same as above, but restricted to multitoken entities
	crunchbase_cased	Gazetteer (case-sensitive) using the Crunchbase Open Data Map
	multitoken_crunchbase_cased	Same as above, but restricted to multitoken entities
	crunchbase_uncased	Gazetteer (case-insensitive) using the Crunchbase Open Data Map
	multitoken_crunchbase_uncased	Same as above, but restricted to multitoken entities
	geo_cased	Gazetteer (case-sensitive) using the Geonames database
	multitoken_geo_cased	Same as above, but restricted to multitoken entities
	geo_uncased	Gazetteer (case-insensitive) using the Geonames database
	multitoken_geo_uncased	Same as above, but restricted to multitoken entities
Heuristic functions	product_cased	Gazetteer (case-sensitive) using products extracted from DBpedia
	multitoken_product_cased	Same as above, but restricted to multitoken entities
	product_uncased	Gazetteer (case-insensitive) using products extracted from DBpedia
	multitoken_product_uncased	Same as above, but restricted to multitoken entities
	date_detector	Detection of entities of type DATE
	time_detector	Detection of entities of type TIME
	money_detector	Detection of entities of type MONEY
	number_detector	Detection of entities CARDINAL, ORDINAL, PERCENT and QUANTITY
	legal_detector	Detection of entities of type LAW
	misc_detector	Detection of entities of type NORP, LANGUAGE, FAC or EVENT
	full_name_detector	Heuristic function to detect full person names
	company_type_detector	Detection of companies with a legal type suffix
	nnp_detector	Detection of sequences of tokens with NNP as POS-tag
	infrequent_nnp_detector	Detection of sequences of tokens with NNP as POS-tag + including at least one infrequent token (rank > 15000 in vocabulary)
	proper_detector	Detection of proper names based on casing
	infrequent_proper_detector	Detection of proper names based on casing + including at least one infrequent token
	proper2_detector	Detection of proper names based on casing
	infrequent_proper2_detector	Detection of proper names based on casing + including at least one infrequent token
	compound_detector	Detection of proper noun phrases with compound dependency relations
	infrequent_compound_detector	Detection of proper noun phrases with compound dependency relations + including at least one infrequent token
	snips	Probabilistic parser specialised in the recognition of dates, times, money amounts, percents, and cardinal/ordinal values
Document-level functions	doc_history	Entity classification based on already introduced entities in the document
	doc_majority_cased	Entity classification based on majority labels in document (case-sensitive)
	doc_majority_uncased	Entity classification based on majority labels in document (case-insensitive)

Table 3: Full list of labelling functions employed in the experiments. The neural NER models are provided in two versions: one that directly outputs the raw model predictions, and one that runs a shallow postprocessing step on the model predictions to correct known recognition errors (for instance, ensuring that a numeric amount that is either preceded or followed by a currency symbol is always classified as an entity of type MONEY).

B Label matching problem

The baseline models relying on mixtures of multinomials have to address the so-called *label matching problem*, which needs some extra care.

The following approach was employed in the experiments from Section 4:

- First, we put strong initial values to the probabilities σ of individual classes based on the frequency of appearance of these classes in the most reliable labelling function. This is expected to increase the probability of EM exploring the mode around the initialised values.
- Second, we perform post-processing and set the labels to the states corresponding to the labels with the highest pairwise correlations to the latent labels from one of the three options:
 1. the most reliable labelling function (Ontonotes-trained NER);
 2. the majority voting labelling function;
 3. the suggested Dirichlet dependent mixture model.

Additionally, if this highest correlation is below the threshold of 0.1 the O label is assigned to the corresponding state. We empirically observed that the label matching technique that performed best was to map the states to the labels produced by the majority voter (based on the pairwise correlations).

C Detailed results

In Table 4, we provide the detailed results distributed by NER label for the CoNLL data 2003 which were presented in micro-averaged form in Table 1 of the main paper.

Label	Proportion	Model	Token-level			Entity-level		
			P	R	F_1	P	R	F_1
LOC	30.3 %	Ontonotes-trained NER	0.767	0.812	0.788	0.764	0.800	0.782
		Majority voting (MV)	0.740	0.839	0.786	0.739	0.828	0.780
		Confusion Matrix	0.721	0.895	0.798	0.714	0.890	0.792
		Sequential Confusion Matrix	0.681	0.856	0.758	0.664	0.848	0.744
		Dependent Confusion Matrix	0.718	0.890	0.794	0.710	0.886	0.788
		Snorkel-aggregated labels	0.634	0.855	0.728	0.676	0.747	0.710
		HMM (only NER models)	0.601	0.825	0.696	0.650	0.733	0.690
		HMM (only gazetteers)	0.707	0.632	0.668	0.694	0.630	0.660
		HMM (heuristics)	0.715	0.870	0.784	0.745	0.832	0.786
		HMM (all but doc-level)	0.701	0.862	0.774	0.724	0.838	0.776
		HMM (all functions)	0.726	0.859	0.786	0.738	0.839	0.786
		NN trained on HMM	0.736	0.851	0.790	0.734	0.850	0.788
PER	28.7 %	Ontonotes-trained NER	0.850	0.833	0.842	0.787	0.741	0.764
		Majority voting (MV)	0.915	0.871	0.892	0.831	0.775	0.802
		Confusion Matrix	0.891	0.921	0.906	0.806	0.834	0.820
		Sequential Confusion Matrix	0.849	0.879	0.864	0.730	0.789	0.758
		Dependent Confusion Matrix	0.892	0.920	0.906	0.806	0.834	0.820
		Snorkel-aggregated labels	0.816	0.903	0.858	0.769	0.717	0.742
		HMM (only NER models)	0.837	0.860	0.848	0.770	0.744	0.756
		HMM (only gazetteers)	0.917	0.452	0.606	0.835	0.391	0.532
		HMM (heuristics)	0.836	0.933	0.882	0.791	0.799	0.794
		HMM (all but doc-level)	0.859	0.917	0.888	0.814	0.782	0.798
		HMM (all functions)	0.857	0.947	0.900	0.820	0.826	0.822
		NN trained on HMM	0.856	0.946	0.898	0.814	0.824	0.818
ORG	26.6 %	Ontonotes-trained NER	0.536	0.517	0.526	0.437	0.306	0.360
		Majority voting (MV)	0.725	0.512	0.600	0.610	0.434	0.508
		Confusion Matrix	0.698	0.613	0.652	0.571	0.537	0.554
		Sequential Confusion Matrix	0.632	0.590	0.610	0.485	0.515	0.500
		Dependent Confusion Matrix	0.696	0.613	0.652	0.567	0.536	0.552
		Snorkel-aggregated labels	0.512	0.639	0.568	0.519	0.496	0.508
		HMM (only NER models)	0.516	0.549	0.532	0.425	0.333	0.374
		HMM (only gazetteers)	0.648	0.304	0.414	0.512	0.235	0.322
		HMM (heuristics)	0.566	0.625	0.594	0.549	0.501	0.524
		HMM (all but doc-level)	0.565	0.631	0.596	0.551	0.494	0.520
		HMM (all functions)	0.542	0.665	0.598	0.545	0.527	0.536
		NN trained on HMM	0.539	0.665	0.596	0.537	0.519	0.528
MISC	14.4 %	Ontonotes-trained NER	0.676	0.599	0.636	0.702	0.583	0.636
		Majority voting (MV)	0.861	0.187	0.308	0.809	0.193	0.312
		Confusion Matrix	0.895	0.319	0.470	0.850	0.332	0.478
		Sequential Confusion Matrix	0.850	0.320	0.464	0.791	0.333	0.468
		Dependent Confusion Matrix	0.893	0.318	0.468	0.844	0.330	0.474
		Snorkel-aggregated labels	0.852	0.398	0.542	0.863	0.400	0.546
		HMM (only NER models)	0.667	0.544	0.600	0.708	0.518	0.598
		HMM (only gazetteers)	0.745	0.011	0.022	0.594	0.008	0.016
		HMM (heuristics)	0.842	0.499	0.626	0.850	0.478	0.612
		HMM (all but doc-level)	0.714	0.596	0.650	0.781	0.575	0.662
		HMM (all functions)	0.814	0.571	0.672	0.830	0.565	0.672
		NN trained on HMM	0.852	0.577	0.688	0.866	0.583	0.696

Table 4: Detailed evaluation results on the CoNLL2003 dataset, depending on NER labels.