A Neural Model for Aggregating Coreference Annotation in Crowdsourcing

Maolin Li[1,2], **Hiroya Takamura**[2,3], **Sophia Ananiadou**[1,2,4]
[1]National Centre for Text Mining, The University of Manchester, Manchester, United Kingdom
[2] Artificial Intelligence Research and Technology,
National Institute of Advanced Industrial Science and Technology, Tokyo, Japan
[3] Precision and Intelligence Laboratory, Tokyo Institute of Technology, Tokyo, Japan
[4] The Alan Turing Institute, London, United Kingdom
{maolin.li, sophia.ananiadou}@manchester.ac.uk
takamura.hiroya@aist.go.jp

Abstract

Coreference resolution is the task of identifying all mentions in a text that refer to the same real-world entity. Collecting sufficient labelled data from expert annotators to train a high-performance coreference resolution system is time-consuming and expensive. Crowdsourcing makes it possible to obtain the required amounts of data rapidly and cost-effectively. However, crowd-sourced labels can be noisy. To ensure high-quality data, it is crucial to infer the correct labels by aggregating the noisy labels. In this paper, we split the aggregation into two subtasks, i.e, mention classification and coreference chain inference. Firstly, we predict the general class of each mention using an autoencoder, which incorporates contextual information about each mention, while at the same time taking into account the mention's annotation complexity and annotators' reliability at different levels. Secondly, to determine the coreference chain of each mention, we use weighted voting which takes into account the learned reliability in the first subtask. Experimental results demonstrate the effectiveness of our method in predicting the correct labels. We also illustrate our model's interpretability through a comprehensive analysis of experimental results.

1 Introduction

Coreference resolution is the task of identifying all mentions in a text that refer to the same real-world entity. However, it is time-consuming and expensive to collect the large amounts of data from expert annotators that are required to train high-performance coreference resolution systems. A rapid and cost-effective alternative is to obtain labels through crowdsourcing (Snow et al., 2008). However, crowd-sourced labels are often noisy. In the example in Figure 1, the mention *it* actually refers to *The Super Lamb Banana*. However, different crowd annotators produced conflicting labels for this mention. We can also observe that the coreference annotation is more complex than classification and sequence labels. Annotators have to determine an appropriate referent mention for some mentions. Because the performance of supervised learning models is highly dependent on the quality of training data, the *aggregation* of these noisy labels, (i.e., the process of determining the label that is most likely to be correct) is important to obtain a high-quality training corpus. Although label aggregation is a well-studied topic, most existing studies of natural language labelling tasks have only focused on aggregating classification or sequence labels. To the best of our knowledge, there is only one previous study (Paun et al., 2018) that has investigated how to aggregate crowd-sourced coreference labels.

In this paper, we propose a 2-step framework in which the aggregation task is broken down into two subtasks, i.e., mention classification and coreference chain inference.

In the mention classification subtask, our model predicts the general category of a mention as shown in Table 1. Our model is based on the autoencoder proposed in (Yin et al., 2017), but with significant extensions. Our encoder is a classifier which takes as its input the crowd labels for each mention, together with the mention's context information. Then it predicts the most plausible general class label for the

Proceedings of the 28th International Conference on Computational Linguistics, pages 5760–5773
Barcelona, Spain (Online), December 8-13, 2020

Text and Gold Answer:

...
The Super Lamb Banana is a bright yellow sculpture located in Liverpool, England.

Weighing almost eight tons and standing at 17 feet tall, it is intended to be a cross between a banana and a lamb and was created by Manhattan-based Japanese artist Taro Chiezo.
...

Crowdsourced Labels:

Annotator 1: *it* → **Discourse Old:** *The Super Lamb Banana*
Annotator 2: *it* → **Discourse Old:** *a bright yellow sculpture located in Liverpool, England*
Annotator 3: *it* → **Discourse New:** the mention is a new entity in the text
...

Figure 1: An example (adapted from the Phrase Detectives Corpus (Chamberlain et al., 2016)) of crowd-sourced coreference annotation.

General Label Type	Description
Discourse New (DN)	The mention is a new entity in the text
Discourse Old (DO)	The mention refers to an entity which has already been introduced
Non-Referring (NR)	The mention refers to no actual entity (e.g., *it* in expletive constructions)
Property (PR)	The mention refers to a property of an entity (e.g., *the most durable light* is a property of *the bulb*)

Table 1: Four general label types in a coreference resolution labelling task (Chamberlain et al., 2016).

mention, by taking into account the annotation complexity of the mention and annotators' reliability. The first challenge in proposing the encoder is how to incorporate mention context information. We explore the use of contextual embeddings for this purpose. The second challenge is how to effectively model annotators' behaviour in terms of their quality of annotation. Modelling annotator reliability is helpful in detecting unreliable annotators and in facilitating appropriate task allocation (Donmez and Carbonell, 2008; Donmez and Carbonell, 2010; Li et al., 2017). Modelling only per-category reliability may not be sufficient to characterise annotators' behaviour patterns for a given annotation task. The original encoder from (Yin et al., 2017) already estimates the per-category reliability. We additionally model overall and per-instance reliability. In addition, we also model the instance complexity.

In the second subtask, i.e. coreference chain inference, based on the predicted general classes in the first subtask, we predict each mention's target (i.e., its referent entity which is usually another mention in the text). If a mention is classified as *Discourse Old* (i.e., it refers to an entity mentioned previously in the text) or as the *Property* of another mention, its coreference chain is inferred using weighted voting in which the annotators' labels are weighted by their reliability.

Our contributions are as follows: a) We propose a simple but efficient two-step framework for aggregating crowd-sourced coreference labels. b) We investigate how information about context, annotator reliability and instance complexity can be incorporated into our encoder network to infer correct labels. c) Experimental results demonstrate that incorporating mention context, annotator reliability and instance complexity can increase the accuracy of correct label prediction. Moreover, we conduct a comprehensive analysis that shows the learned complexity and reliability are explainable.

2 Related Work

There have been many released coreference corpora in which the annotations were collected from a small number of in-house annotators such as experts (Sundheim, 1995; Hirschman and Chinchor, 1998; Bagga and Baldwin, 1999; Doddington et al., 2004; Pradhan et al., 2012; Singh et al., 2012; Guillou et al., 2014; Garcia and Gamallo, 2014; Chaimongkol et al., 2014; Ghaddar and Langlais, 2016; Cohen et al., 2017; Fonseca et al., 2017; Webster et al., 2018; Bamman et al., 2020; Tsvetkova, 2020). These annotators were assumed to be reliable. Guha et al. (2015) and Chamberlain et al. (2016) attempted to collect coreference annotations from non-expert crowd annotators. Even though crowd aggregation has been studied for many years, most existing studies have focused on aggregating classification labels (Dawid and Skene, 1979; Snow et al., 2008; Raykar et al., 2010; Hovy et al., 2013; Li et al., 2014; Felt et al., 2015; Zheng et al., 2017; Yin et al., 2017; Rodrigues and Pereira, 2018; Guan et al., 2018; Li et al., 2019; Zhang et al., 2019) or sequence labels (Hovy et al., 2014; Rodrigues et al., 2014; Huang et al., 2015; Nguyen et al., 2017; Nye et al., 2018; Yang et al., 2018; Lin et al., 2019). Note that Raykar et al. (2010) and Felt et al. (2015) also included contextual information. Raykar et al. (2010) incorporated a classifier into their Bayesian model. The classifier took an instance's representation as its input and then predicted an answer. The unsupervised Latent Dirichlet Allocation topic model (Blei et al., 2003) which can capture topics of words and documents was extended by Felt et al. (2015) to be able to handle crowdsourced noisy labels. However, these models can not be applied to the coreference annotation in a straightforward manner, because coreference labels are more complex and they are very different from

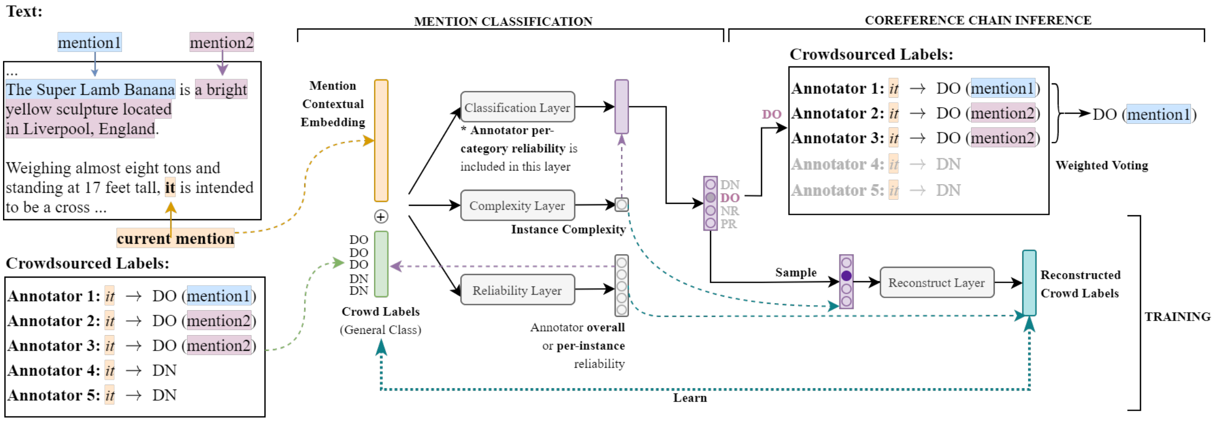

Figure 2: Overview of the proposed method.

classification and sequence labels.

To overcome the lack of a suitable aggregation method for coreference annotations, Paun et al. (2018) proposed the Mention-Pair Annotation model, which to the best of our knowledge, is the only study that attempts to address this challenge. They defined a graphical model and introduced a true label indicator for each <General Class, referent mention> pair, indicating whether or not the pair is correct. However, this model does not include contextual information, which may support the prediction of the correct labels, because the meaning of mentions usually depends on the context in which they occur.

3 Model

We break down the aggregation of crowd-sourced coreference labels into two steps: **mention classification** and **coreference chain inference**, as illustrated in Figure 2. Below, we describe our method in more detail. All the biases in linear layer parameters are omitted for simplification. Table 2 contains the main notations of our model.

Notation	Description
N	Number of instances (i.e., mentions that have crowd-sourced labels) in a dataset
T	Number of annotators
K	Number of general classes
$\mathbf{C}^n$	Crowd general labels of n-th mention (a $T \times K$ matrix)
$\mathbf{c}^n$	Flattened $\mathbf{C}^n$
$\tilde{\mathbf{C}}^n$	Reconstructed $\mathbf{C}^n$ by a decoder network
$\mathbf{x}^n$	Contextual embedding of n-th mention
d^n	Annotation complexity of n-th mention
$\mathbf{r}_{ct}$	Per-category reliability of t-th annotator (a $K \times K$ matrix)
r_{ot}	Overall reliability of t-th annotator (a scalar)
r_{nt}	Per-instance reliability of t-th annotator on n-th mention (a scalar)
w_{to}, $\mathbf{w}_r^t$	Learnable parameters for computing the t-th annotator's overall and per-instance reliability respectively
$\mathbf{w}_f$	Learnable parameters for computing instance complexity
$\mathbf{W}_e$, $\mathbf{W}_d$	Learnable parameters of the encoder and decoder network respectively

Table 2: Main notation for our proposed models.

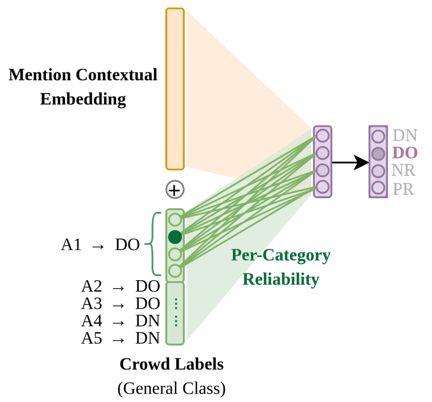

Figure 3: Per-category reliability in the method. Each annotator's label is encoded as a vector which is described in Section 3.1.1.

3.1 Mention Classification

In this step, the encoder network, which is a feed-forward neural network classifier, receives as input a mention's contextual embedding and crowd-sourced labels weighted by the annotators' reliability. The output of this encoder is the mention's predicted general class (i.e., DN, DO, NR, or PR).

To prepare the input, we first obtain each mention's contextual representation from a pre-trained embedding model, and then concatenate the mention's crowd labels with its contextual representation. This concatenated vector is used as input to the complexity layer, which computes the instance complexity.

The output of the complexity layer is used to weight the encoder output, which indicates how much attention our model should pay to learn this instance. The goal is for our model to automatically pay more attention to learning difficult instances than easy ones. Next, we use the mention's contextual embedding to compute annotator reliability, in order to weight the crowd-sourced labels in the concatenated vector. Finally, the weighted concatenated vector is considered as the final input of our encoder.

Since the ground truth is unavailable in real-world situations, the expert labels in the dataset cannot be used for training the encoder. We introduce a decoder network, which is another neural network that reconstructs the encoder input. By using the reconstructed input, we maximise the log-likelihood of all the observed crowd-sourced labels in the dataset to train the entire autoencoder.

3.1.1 Crowd Labels and Mention Contextual Embedding

Crowd Labels: For the n-th mention, we formulate its crowd-sourced general labels as a $T \times K$ matrix $\mathbf{C}^n$. C^n_{tk} is set to 1 if the t-th annotator's label is the k-th general label, otherwise to 0.

Mention Contextual Embedding: Since a mention could consist of more than one token , the average of each token's pre-trained contextual embedding is taken as the representation of the mention, which is represented by the symbol $\mathbf{x}^n$.

3.1.2 Instance Complexity

The complexity of annotating a certain instance is an important factor affecting the quality of annotations produced by crowd workers. If a given mention is more challenging to annotate than others, the annotators are likely to need more effort and time to assign a label for this mention. This is in contrast to assigning labels for easier instances. We aim for our model to perform similarly to annotators who pay more attention to more challenging instances than easier ones.

More specifically, we assume that the complexity of annotating the n-th instance, d^n, can be estimated from an instance $\mathbf{x}^n$ and its corresponding crowd labels $\mathbf{c}^n$. Therefore, d^n is computed as:

$$d^n = softplus([\mathbf{x}^n; \mathbf{c}^n] \cdot \mathbf{w}_f). \tag{1}$$

We concatenate the flattened $\mathbf{C}^n$ represented by $\mathbf{c}^n$ with $\mathbf{x}^n$. Then, we use the softplus activation function to compute the annotation complexity of the n-th instance. $\mathbf{w}_f$ is the parameter vector which will be learned during training.

3.1.3 Annotator Reliability

Apart from modelling the per-category reliability of annotators, we introduce an additional layer to estimate the overall and instance-level reliability.

t-th Annotator's Per-Category Reliability $\mathbf{r}_{ct}$: We consider the encoder network parameters corresponding to the crowd label as the per-category reliability, as illustrated in Figure 3. For each annotator, these parameters can be reshaped into a $K \times K$ confusion matrix. The parameter located in the i-th row and the j-th column indicates the extent to which an annotator prefers to assign the j-th category if the true answer is the i-th category. If the matrix is nearer to a diagonal matrix with positive values, it means that this annotator can reliably annotate all categories. The parameters will be learned during training.

t-th Annotator's Overall Reliability r_{ot}: For the overall score, we assign a scalar w_{to}, which will be learned during training, following a sigmoid function to the t-th annotator as:

$$r_{ot} = (1 + e^{-w_{to}})^{-1}. \tag{2}$$

t-th Annotator's Per-Instance Reliability r^n_t: For the per-instance reliability, we use $\mathbf{x}^n$ and a sigmoid function to compute the t-th annotator's reliability on the n-th instance:

$$r^n_t = (1 + e^{-(\mathbf{x}^n \cdot \mathbf{w}^t_r)})^{-1}. \tag{3}$$

$\mathbf{w}^t_r$ is the parameter vector which will be learned during training.

3.1.4 Encoder Network

The encoder maps its input to a probability distribution $p(y^n|\mathbf{x}^n, \mathbf{C}^n)$ over the set of general classes, where $y \in \{DN, DO, NR, PR\}$. We prepare the encoder input for the n-th mention in the following way. Firstly, the crowd label matrix $\mathbf{C}^n$ and mention contextual representation $\mathbf{x}^n$ are obtained as described in Section 3.1.1. Secondly, we weigh each annotator's label by multiplying each row of $\mathbf{C}^n$ (whose values correspond to a given annotator's labels) by the reliability score of the annotator. The weighted matrix is denoted by the symbol $\mathbf{C}^{n\prime}$.

Specifically, since the encoder already includes the per-category reliability, which will weight crowd-sourced labels (described in Section 3.1.3), we let $\mathbf{C}^{n\prime}$ be $\mathbf{C}^n$ when using only this reliability:

$$\mathbf{C}^{n\prime} = \mathbf{C}^n. \tag{4}$$

When per-category reliability is additionally supplemented by overall reliability (computed by Equation (2)), we compute $\mathbf{C}^{n\prime}$ as:

$$\mathbf{C}^{n\prime} = [r_{o1}, r_{o2}, ..., r_{oT}]^{\mathrm{T}} \odot \mathbf{C}^n. \tag{5}$$

$\odot$ is the element-wise multiplication and T is the number of annotators, while T means transpose of a matrix.

When supplementing the per-category reliability with the per-instance reliability, we firstly estimate each annotator's reliability score on each instance r_t^n using Equation (3), and then compute $\mathbf{C}^{n\prime}$ as:

$$\mathbf{C}^{n\prime} = [r_1^n, r_2^n, ..., r_T^n]^{\mathrm{T}} \odot \mathbf{C}^n. \tag{6}$$

Finally, we flatten $\mathbf{C}^{n\prime}$ to a vector $\mathbf{c}^{n\prime}$ and implement the encoder as:

$$p(y^n|\mathbf{x}^n, \mathbf{C}^n) = \text{softmax}(d^n \odot ([\mathbf{c}^{n\prime}; \mathbf{x}^n] \times \mathbf{W}_e)), \tag{7}$$

where the complexity d^n is computed using Equation (1) and $\mathbf{W}_e$ is the learnable encoder parameter.

3.1.5 Decoder Network

The decoder is another network which reconstructs the input crowd annotations[1] $\mathbf{C}^n$. We firstly sample a label $\mathbf{y}$ from the predicted general class distribution $p(y^n|\mathbf{x}^n, \mathbf{C}^n)$ provided by the encoder. $\mathbf{y}$ is a one-hot encoding. We then compute the reconstructed crowd labels $\tilde{\mathbf{C}}^n$ as:

$$\tilde{\mathbf{v}}^n = d^n \odot (\mathbf{y} \times \mathbf{W}_d). \tag{8}$$

$$\tilde{\mathbf{C}}^{n\prime} = \begin{bmatrix} \text{softmax}(\tilde{\mathbf{v}}_{1,K}^{n(1)}) \\ \vdots \\ \text{softmax}(\tilde{\mathbf{v}}_{1,K}^{n(t)}) \\ \vdots \\ \text{softmax}(\tilde{\mathbf{v}}_{1,K}^{n(T)}) \end{bmatrix}, \quad \tilde{\mathbf{v}}_{1,K}^{n(t)} = \tilde{\mathbf{v}}_{(t-1)\times K+1, tK}^n. \tag{9}$$

$$\tilde{\mathbf{C}}^n = \mathbf{r} \odot \tilde{\mathbf{C}}^{n\prime}. \tag{10}$$

$\mathbf{W}_d$ is the learnable decoder parameter, $\tilde{\mathbf{v}}^n$ is the decoder output before application of the activation function (see Equation (8)) and $\tilde{\mathbf{v}}_{1,K}^{n(t)}$ are the K elements from index $(t-1) \times K + 1$ to tK, which correspond to the t-th annotator's reconstructed crowd label for the n-th mention. We apply the annotator-wise softmax function to $\tilde{\mathbf{v}}^n$ as illustrated in Equation (9). Finally, using Equation (10), the annotator reliability $\mathbf{r}$ is used to weight $\tilde{\mathbf{C}}^{n\prime}$ in the same manner as was described above using Equations (4)-(6).

3.1.6 Learning and Predicting

Pre-Training: We firstly pre-train the encoder by using the majority voting labels as targets. This solves a potential problem of using the encoder, i.e., that the meaning of elements in the encoder output vector is exchangeable. Pre-training the encoder can instead make the output vector aware of which element should represent what category.

Training: To train the autoencoder, we maximise the lower bound of log-likelihood of the observed

[1]The reason why we only reconstruct crowd labels instead of also reconstructing mention context is explained in Section 6.3.

data $\mathbf{x}$ and $\mathbf{C}$:

$$\begin{aligned}\log p(\mathbf{x}, \mathbf{C}) &= \sum_{n}^{N} \sum_{y \in \text{classes}} p(y|\mathbf{x}^n, \mathbf{C}^n) \log \frac{p(y)p(\tilde{\mathbf{x}}^n, \tilde{\mathbf{C}}^n|y)}{p(y|\mathbf{x}^n, \mathbf{C}^n)} + D_{KL}(p(y|\mathbf{x}^n, \mathbf{C}^n)||p_{true}(y|\mathbf{x}^n, \mathbf{C}^n)) \\ &\geq \sum_{n}^{N} \sum_{y \in \text{classes}} p(y|\mathbf{x}^n, \mathbf{C}^n) \log \frac{p(y)p(\tilde{\mathbf{x}}^n, \tilde{\mathbf{C}}^n|y)}{p(y|\mathbf{x}^n, \mathbf{C}^n)} \\ &= \sum_{n}^{N} \mathbb{E}_{p(y|\mathbf{x}^n, \mathbf{C}^n)} \log p(\tilde{\mathbf{x}}^n, \tilde{\mathbf{C}}^n|y) - \lambda_1 D_{KL}(p(y|\mathbf{x}^n, \mathbf{C}^n)||p(y)),\end{aligned} \tag{11}$$

where $p_{true}(y|\mathbf{x}^n, \mathbf{C}^n)$ is the true distribution (which is unknown) and D_{KL} is a Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951), which measures the difference between probability distributions. The prior probability $p(y)$ in the KL term is estimated from the labels predicted using majority voting. During learning, the KL divergence between the encoder prediction $p(y|\mathbf{x}^n, \mathbf{C}^n)$ and the prior $p(y)$ ensures that our model retains an awareness of the category position information which is learned in the pre-training step. We also introduce a strength hyper-parameter λ_1 on the KL term to weight the impact of $p(y)$ on learning the encoder. Note that in this equation, the decoder $p(\tilde{\mathbf{x}}^n, \tilde{\mathbf{C}}^n|y)$ reconstructs both the mention representation $\mathbf{x}^n$ and the crowd labels $\mathbf{C}^n$. However, since we found that this model not only runs more slowly, but also obtains lower performance than the model which only reconstructs crowd labels, we decided not to reconstruct the mention contextual embedding.[2] Therefore, the reconstruction probability $p(\tilde{\mathbf{x}}^n, \tilde{\mathbf{C}}^n|y)$ is computed as:

$$\log p(\tilde{\mathbf{x}^n}, \tilde{\mathbf{C}}^n|y) \propto \log p(\tilde{\mathbf{C}}^n|y) = \sum_{t}^{T} \sum_{k}^{K} C_{tk}^n \log \tilde{C}_{tk}^n. \tag{12}$$

Regularisation: To prevent overfitting, we apply $L1$ and $L2$ regularisation to parameters as:

$$\lambda_2(||\mathbf{w}_r||_1 + ||\mathbf{W}_e||_1 + ||\mathbf{W}_d||_1) + \lambda_3(||\mathbf{w}_f||_2) \tag{13}$$

λ_2 and λ_3 are also strength hyper-parameters.

Prediction: To infer the correct general class, we take each mention's general class as the most probable label according to the distribution $p(y|\mathbf{x}^n, \mathbf{C}^n)$ predicted by the encoder.

3.2 Coreference Chain Inference

In this step, mentions classified as either DN or NR by the encoder are not processed further, as they do not refer to any other mentions in text. For each of the other mentions, we filter out the crowd-sourced DN and NR labels, and apply weighted voting to the remaining crowd labels to infer its referent mention. In other words, for a mention which is classified as DO or PR, we only aggregate those crowd-sourced referent mentions that appear in the set of mentions classified as DO($\cdot$) or PR($\cdot$)[3] labels. Each annotator's label is weighted by the product of the annotator's category and the overall/per-instance reliability.

4 Experiments

Dataset: We evaluate our method on the real-world dataset from (Chamberlain et al., 2016), which includes both crowd labels produced by 280 crowd workers and expert labels for 5,654 mentions (3,277 DNs, 2,192 DOs, 136 PRs and 49 NRs).

Mention Contextual Embedding: We compare the use of two pre-trained embeddings from ELMo[4] (Peters et al., 2018) and BERT (bert-base-uncased)[5] (Devlin et al., 2019). When using BERT, we represent each token by using the BERT model outputs from the last four hidden layers, which is the same setting as used in (Peters et al., 2018).

Learning: We use the Adam (Kingma and Ba, 2015) optimiser ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$). λ_1, λ_2 and λ_3 are set to 0.0001, 0.005 and 0.5, respectively. We pre-train the encoder for 100 epochs.

[2]The details are discussed in Section 6.3.

[3]For example, DO(*mention1*) or PR(*mention1*) indicates that this annotator labels the current mention as referring to another mention *mention1*.

[4]Original(5.5B): `https://allennlp.org/elmo`

[5]We used the tool developed by (Wolf et al., 2019) to extract BERT embeddings.

With Singletons	MUC			B-cubed			CEAFe			CoNLL
	P	R	F	P	R	F	P	R	F	Score
Majority Voting	**95.18**	69.44	80.30	**95.53**	78.79	86.36	79.04	95.12	86.34	84.33
Mention-Pair Annotation Model (Paun et al., 2018)	92.87	86.07	89.34	94.79	88.56	91.57	90.53	94.27	92.36	91.09
Ours - Without Context, Per-Category Reliability	92.11	94.20	93.14	93.31	93.10	93.20	96.86	95.42	96.13	94.16
- ELMo, Per-Category Reliability	91.87	95.88	93.83	93.95	93.60	93.77	97.65	95.51	96.57	94.72
- BERT, Per-Category Reliability	92.42	95.80	94.08	93.96	93.02	93.49	97.74	96.19	96.96	94.84
- ELMo, Per-Category + Overall Reliability	92.51	95.59	94.02	94.28	93.72	94.00	97.78	95.85	96.81	94.94
- BERT, Per-Category + Overall Reliability	93.42	96.26	**94.82**	95.03	93.78	94.40	97.93	96.83	97.38	95.53
- ELMo, Per-Category + Per-Instance Reliability	92.33	95.99	94.12	94.08	**94.02**	94.05	97.65	95.73	96.68	94.95
- BERT, Per-Category + Per-Instance Reliability	93.21	**96.34**	94.75	95.07	93.79	**94.43**	**97.98**	**97.11**	**97.54**	**95.57**

Without Singletons	MUC			B-cubed			CEAFe			CoNLL
	P	R	F	P	R	F	P	R	F	Score
Majority Voting	**95.18**	69.44	80.30	**93.36**	46.05	61.68	64.23	55.17	59.35	67.11
Mention-Pair Annotation Model (Paun et al., 2018)	92.87	86.07	89.34	88.46	72.83	79.89	79.65	76.32	77.95	82.39
Ours - Without Context, Per-Category Reliability	92.11	94.20	93.14	85.13	83.98	84.55	79.61	80.80	80.20	85.96
- ELMo, Per-Category Reliability	91.87	95.88	93.83	85.68	85.50	85.59	80.03	81.46	80.74	86.72
- BERT, Per-Category Reliability	92.42	95.80	94.08	85.92	84.27	85.09	81.28	81.81	81.54	86.90
- ELMo, Per-Category + Overall Reliability	92.51	95.77	94.11	86.33	86.10	86.21	81.66	81.46	81.56	87.30
- BERT, Per-Category + Overall Reliability	93.42	96.26	**94.82**	88.25	86.27	87.25	83.18	83.62	83.40	88.49
- ELMo, Per-Category + Per-Instance Reliability	92.33	95.99	94.12	86.13	86.65	86.39	80.92	81.96	81.44	87.32
- BERT, Per-Category + Per-Instance Reliability	93.21	**96.34**	94.75	88.19	**87.14**	**87.66**	**83.38**	**84.42**	**83.90**	**88.77**

Table 3: Precision (P), Recall (R) and F scores (F) of our predicted labels.

Complexity	Instances with Lowest Complexities
6.98e-23	The Metric Marvels is **a series of seven animated educational shorts featuring songs about meters, liters, Celsius, and grams, designed to teach American children how to use the metric system**.
1.93e-22	An encounter with German tourists in New Zealand led to the formation of **a group called ”Extreme Ironing International”**, and the German Extreme Ironing Section or GEIS.
2.01e-22	Taro Tsujimoto is **an imaginary ice hockey player that was legally drafted by the National Hockey League's Buffalo Sabres in the 11th round of the 1974 NHL Entry Draft**.
Complexity	**Instances with Highest Complexities**
0.7123	... and so she ran from the path into **the wood** to look for flowers ...
0.7150	Next they came to **some fine meadows**.
0.7188	'Pull off my boots,' and then he threw them in **her face**, and made her pick them up again, and clean and brighten them.

Table 4: Instances with lowest and highest complexities as estimated by our model. Mentions are highlighted in bold.

We then run the entire autoencoder training by optimising the objective function in Equation (11) until either 300 iterations are reached, or the objective function stops improving.

Evaluation: The baselines are: majority voting and the state-of-the-art method, Mention-Pair Annotation model (Paun et al., 2018). Four metrics are used for evaluation, MUC (Vilain et al., 1995), B-cubed (Bagga and Baldwin, 1998), CEAFe (Luo, 2005), and CoNLL Score (Pradhan et al., 2011).

5 Results

From Table 3, we observe that our method achieved better performance than baselines.[6] We also report the performance using different settings. ***Without Context*** and ***ELMo/BERT*** denote the models that do not use context, or which use context, respectively. In terms of reliability, each annotator's ***Per-Category Reliability*** is modelled in our model by default. ***Per-Category + Overall / Per-Instance Reliability*** indicates that the model supplements per-category reliability with modelling of annotator overall or per-instance reliability. As shown in Table 3, the model ***(ELMo/BERT, Per-Category Reliability)*** outperforms ***(Without Context, Per-Category Reliability)***, which suggests that the incorporation of context helps to improve the performance. We note that the performance also benefits from additionally capturing annotators' ***overall*** or ***per-instance*** reliability.

6 Analysis and Discussion

6.1 Instance Complexity

To analyse complexities, we rank instances according to their complexities estimated by Equation (1). Table 4 lists the instances with the lowest and highest complexities. It can be observed that: 1) it is very easy to annotate instances which have low complexities. The meaning of these mentions in text is clear and explicit; 2) It seems that the instances with the highest complexities are short mentions, particularly those containing possessive pronouns or determiners. They are more difficult because the annotator is likely to have to look back to previous sentences to determine whether they are referring to an entity (or a property of an entity) that has previously been introduced. 3) Our model places more emphasis on learning more difficult instances than easier ones.

[6]Mentions that appear only once are *singletons*. The MUC scores with and without singletons are the same because it is not sensitive to singletons (Kübler and Zhekova, 2011).

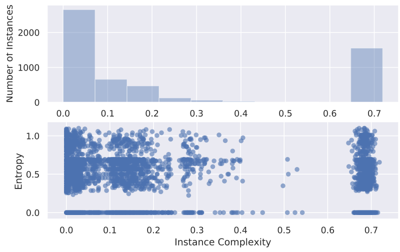

Figure 4: Levels of complexity distributions of instances in the corpus (Top) and correlation between complexity and annotation agreement (Bottom).

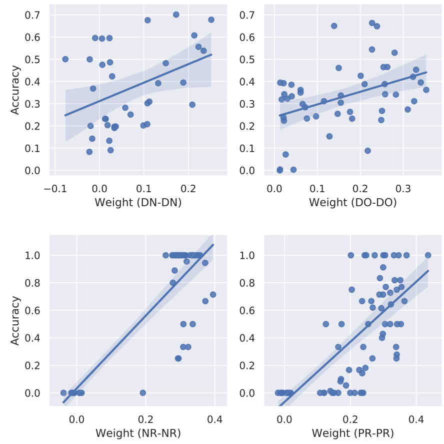

Figure 5: Correlation between annotator's true accuracy with the weight value located in the i-th row and i-th column in annotator's per-category reliability matrix. Each point represents one annotator.

We also investigate the complexity distributions of the corpus and the correlation between each instance's complexity and annotation agreement in Figure 4. From the distribution (top of Figure 4) we can see that the model can distinguish between those instances that are useful for training and those that are not useful. To ascertain whether annotators achieve higher levels of agreement on less challenging instances, we use entropy to measure whether the annotators can make the same annotation decision for a certain instance. A low entropy value indicates a high agreement, and vice versa.

$$\text{Agreement}(\mathbf{a}_i) = -\sum_{k}^{K} P_i(k) \log P_i(k), P_i(k) = \frac{\sum_t^T I(a_{it} = k)}{|\mathbf{a}_i|} \tag{14}$$

where $\mathbf{a}_i$ denotes the annotations for the i-th instance and a_{it} is the t-th annotator's label. $P_i(k)$ is the probability that the i-th instance is annotated as the k-th class. $|\mathbf{a}_i|$ indicates how many annotators annotated this instance. The indication function $I(\cdot) = 1$ when $a_{it} = k$, otherwise $I(\cdot) = 0$.

Figure 4 (lower part) shows no correlation between complexity and agreements, indicating that it is not accurate to measure complexity by relying solely on annotators' levels of agreements.

6.2 Annotator Reliability

6.2.1 Per-Category Reliability

As described in Section 3.1.3, we consider an annotator's per-category reliability as a $K \times K$ confusion matrix (K is the number of classes). The value located in the i-th row and the j-th column indicates the extent to which an annotator prefers to assign the j-th category if the true answer is the i-th category. To explore whether the value located in the i-th row and the i-th column can reflect each annotator's reliability for the i-th category, we visualise annotators' true accuracy of the i-th category and the weight value in Figure 5.[7] We observe that the correlation coefficient of each line is positive, i.e., higher per-category reliability is correlated with higher accuracy. We also present the per-category reliability matrices of six randomly selected annotators in Figure 6. The values in the diagonals of Annotator 0 and 9 are relatively large, indicating that they are both skilled at labelling all the categories. Meanwhile the matrix of Annotator 11 shows that this annotator is good at every category except NR. The matrices of Annotator 41, 57 and 85 suggest they are less reliable than the other annotators.

6.2.2 Per-Instance Reliability

To explore the per-instance reliability, we reduce each instance's embedding representation to a 2-dimensional space for visualisation by using t-SNE (Maaten and Hinton, 2008) as shown in Figure 7. Each point is a single instance and is coloured according to each annotator's estimated reliability on this

[7]We omit the annotators with zero weights for brevity as their labels are recognised as redundant annotations.

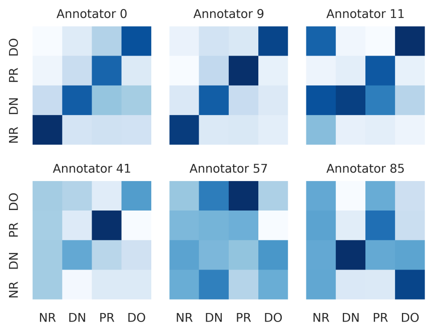

Figure 6: Reliability matrix learned by our model. Darker colours denote higher weights.

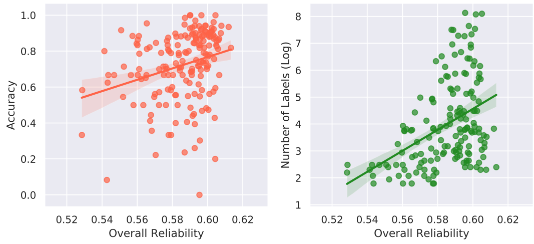

Figure 8: Correlation between annotator's overall reliability with accuracy and number of annotated labels. Each point represents an individual annotator.

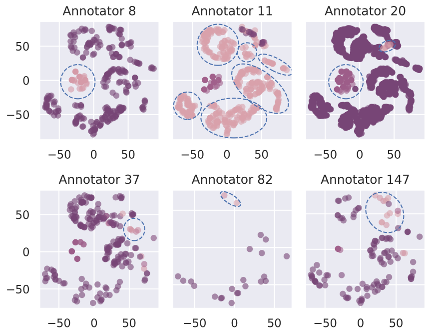

Figure 7: T-SNE visualisation of instances which are annotated by different annotators.

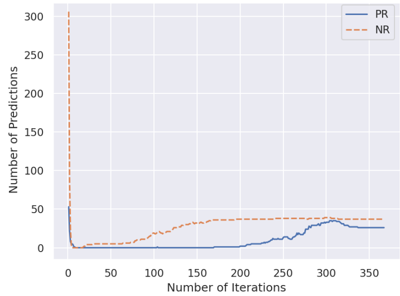

Figure 9: Number of PR and NR predictions after each training iteration.

particular instance. Darker colours mean higher reliability values. Regions where instances have low values enclosed using dotted lines. Note that each figure has a different number of data points, because the number of annotated instances varies among different annotators. We observe that each annotator has noticeably different reliabilities across different instances. In particular, the diagrams show that the annotations of Annotators 8 and 11 can complement each other.

6.2.3 Overall Reliability

To analyse the overall reliability, we investigate the correlation between a) overall reliability and annotator's accuracy (left side of Figure 8) and b) overall reliability and the number of instances annotated (right side of Figure 8). Figure 8 shows positive correlations in both cases. This implies that when an annotator's overall reliability is high, then the true overall accuracy and the number of labels provided are large. Our model considers an annotator as generally as more reliable and experienced if this annotator has a higher accuracy and has annotated a great number of instances.

6.3 Model

In addition, we conduct ablation analysis on our models, as shown in Table 5.

Without Pre-Training: For this experiment, we omit the pre-training step described in Section 3.1.6 and directly train the whole autoencoder. The significant performance drop shown in Table 5 indicates that training the entire autoencoder from scratch produces rather poor results.

Reconstruct Both Crowd Labels and Mention Embedding: We also investigate the performance when the decoder reconstructs both crowd and mention embedding. This reconstruction results in a slight drop in performance. We also found that the training takes much longer. Therefore, we recommend that the decoder should only reconstruct crowd labels.

2-Layer Autoencoder: We increase the number of encoder/decoder layers from one to two, which results in a dramatic performance decrease. It suggests that it is not necessary to use deep models.

Wider Context (Window Size=3): We investigate whether the performance can benefit from combining integration of wider contextual information. We concatenate the mention's contextual embedding

With Singletons	**MUC**			**B-cubed**			**CEAFe**			**CoNLL**
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**Score**
Ours - BERT, Per-Category + Per-Instance Reliability	93.21	**96.34**	94.75	95.07	93.79	**94.43**	**97.98**	**97.11**	**97.54**	**95.57**
- Without Pre-Training	**93.42**	3.28	6.33	**99.58**	65.26	78.85	60.94	93.11	73.67	52.95
- Reconstruct Both Crowd Labels and Mention Embedding	91.74	95.38	93.53	93.92	93.87	93.90	97.38	95.25	96.30	94.58
- 2-Layer Autoencoder	91.06	81.32	85.92	94.57	84.71	89.37	88.60	94.41	91.41	88.90
- Wider Context (Window Size=3)	91.95	95.56	93.72	93.28	93.69	93.48	97.46	95.33	96.38	94.53

Without Singletons	**MUC**			**B-cubed**			**CEAFe**			**CoNLL**
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**Score**
Ours - BERT, Per-Category + Per-Instance Reliability	93.21	**96.34**	94.75	88.19	**87.14**	**87.66**	**86.88**	**91.92**	**89.33**	**90.58**
- Without Pre-Training	**93.42**	3.28	6.33	**94.23**	1.81	3.55	40.27	3.74	6.84	5.57
- Reconstruct Both Crowd Labels and Mention Embedding	91.74	95.38	93.53	84.54	86.46	86.00	83.52	88.54	85.96	88.49
- 2-Layer Autoencoder	91.06	81.32	85.92	85.42	62.64	72.27	73.84	68.30	70.97	76.39
- Wider Context (Window Size=3)	91.95	95.56	93.72	84.33	86.11	85.21	83.90	88.24	86.02	88.31

Table 5: Ablation analysis of our method in predicting correct labels.

with the averaged BERT embeddings of the three tokens before and after the current mention. However we find that this does not further improve the performance of the model.

In order to better understand the behaviour of our model, the prediction errors are analysed. In the first step, the general class of a mention can be mistakenly classified. For example, the model wrongly predicted some DO mentions as DN mentions (DO→DN). In the second step, the incorrect coreference chain of a mention can be determined. We found that 79.25% (DO→DN 44.61%, PR→DN 16.18%, DN→DO 12.91%, Others 5.5%) and 20.75% of total errors were made in the first and second step respectively. After checking the mentions with wrong predictions, we summarise the possible reasons as follows: 1) The lack of contextual information from earlier sentences. For example, *the bottle* in the sentence "[...] and break *the bottle*, and [...]" was incorrectly predicted as a DN mention. This bottle had actually been introduced in an earlier sentence. 2) The difficulty of distinguishing between mentions which have closed meaning but belong to different types. Here are two examples: *a wicked creature* which should be considered as a property of *a wolf* instead of a new mention; The *cakes* in "[...] was again taking *cakes* to [...]" was predicted as a DO mention referring to another *cake*. However, it is actually a DN mention because the previously mentioned *cake* had been consumed and the *cakes* are new ones. 3) The challenge of identifying if the mention *it*, is referring to a thing previously mentioned or is in an expletive construction (e.g., the *it* in "How dark *it* was inside the wolf."). Incorporating information from a mention's neighbouring sentences may improve the model. Since there are not many PR and NR mentions in the training data, to investigate how well our model learned them, we checked their numbers of predictions after each iteration as shown in Figure 9. We can observe that the numbers are very close to 0 at first iterations and eventually become stable somewhere between 25 to 40. We also found that approximately 20% of PR predictions were wrong and 75% were not identified. Therefore, it is worth investigating an appropriate training method to deal with the imbalance in training data.

7 Conclusion and Future Work

We proposed a two-step framework for aggregating crowd-sourced coreference labels. In the mention classification subtask, the encoder classifies each mention as belonging to one of the four general categories, i.e., DN, DO, NR or PR. This encoder incorporates mention context, instance complexity and the annotator reliability at different levels (i.e., overall, per-category and per-instance). In the coreference chain inference subtask, we use the learned reliability to infer the coreference chains. Experimental results demonstrate the effectiveness of our model. Furthermore, our comprehensive analysis shows that the learned complexity and reliability are explainable, thus helping to explain how our model infers the correct label for each instance. Lastly, an error analysis was carried out to understand the incorrect predictions. As future work, we will explore the challenges of solving other complex annotation tasks and how our model can be used and adapted for them.

Acknowledgements

This research was supported by BBSRC Japan Partnering Award [Grant ID: BB/P025684/1] and by funding from the Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Japan. We would like to thank Paul Thompson for his valuable comments. M. Li thanks The University of Manchester for the School of Computer Science Kilburn Overseas Fees Bursary.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566, Granada, Spain.

Amit Bagga and Breck Baldwin. 1999. Cross-document event coreference: Annotations, experiments, and observations. In *Coreference and Its Applications*.

David Bamman, Olivia Lewke, and Anya Mansoor. 2020. An annotated dataset of coreference in English literature. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 44–54, Marseille, France, May. European Language Resources Association.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Panot Chaimongkol, Akiko Aizawa, and Yuka Tateisi. 2014. Corpus for coreference resolution on scientific papers. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 3187–3190, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. 2016. Phrase detectives corpus 1.0 crowdsourced anaphoric coreference. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the 10h International Conference on Language Resources and Evaluation (LREC)*, Paris, France, May. European Language Resources Association (ELRA).

K Bretonnel Cohen, Arrick Lanfranchi, Miji Joo-young Choi, Michael Bada, William A Baumgartner, Natalya Panteleyeva, Karin Verspoor, Martha Palmer, and Lawrence E Hunter. 2017. Coreference annotation and resolution in the colorado richly annotated full text (craft) corpus of biomedical journal articles. *BMC Bioinformatics*, 18(1):1–14.

Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, pages 20–28.

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) (NAACL)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May. European Language Resources Association (ELRA).

Pinar Donmez and Jaime G. Carbonell. 2008. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM)*, CIKM '08, pages 619–628, New York, NY, USA. Association for Computing Machinery.

Pinar Donmez and Jaime G. Carbonell. 2010. From active to proactive learning methods. In Jacek Koronacki, Zbigniew W. Raś, Sławomir T. Wierzchoń, and Janusz Kacprzyk, editors, *Advances in Machine Learning I: Dedicated to the Memory of Professor Ryszard S.Michalski*, pages 97–120. Springer Berlin Heidelberg, Berlin, Heidelberg.

Paul Felt, Eric Ringger, Jordan Boyd-Graber, and Kevin Seppi. 2015. Making the most of crowdsourced document annotations: Confused supervised LDA. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 194–203, Beijing, China, July. Association for Computational Linguistics.

Evandro Brasil Fonseca, Sandra Collovini de Abreu, Vinicius Sesti, Ana Luisa Leal, Paulo Quaresma, and Renata Vieira. 2017. Collective elaboration of a coreference annotated corpus for portuguese texts. In *Proceedings of the Second Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval)*, Murcia, Spain, September.

Marcos Garcia and Pablo Gamallo. 2014. Multilingual corpora with coreferential annotation of person entities. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Abbas Ghaddar and Phillippe Langlais. 2016. WikiCoref: An English coreference-annotated corpus of Wikipedia articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pages 136–142, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Melody Guan, Varun Gulshan, Andrew Dai, and Geoffrey Hinton. 2018. Who said what: Modeling individual labelers improves classification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, Louisiana, USA, February.

Anupam Guha, Mohit Iyyer, Danny Bouman, and Jordan Boyd-Graber. 2015. Removing the training wheels: A coreference dataset that entertains humans and challenges computers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1108–1118, Denver, Colorado, USA. Association for Computational Linguistics.

Liane Guillou, Christian Hardmeier, Aaron Smith, Jörg Tiedemann, and Bonnie Webber. 2014. ParCor 1.0: A parallel pronoun-coreference corpus to support statistical MT. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 3191–3198, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Lynette Hirschman and Nancy Chinchor. 1998. Appendix F: MUC-7 coreference task definition (version 3.0). In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

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 (NAACL-HLT)*, pages 1120–1130, Atlanta, Georgia, June. Association for Computational Linguistics.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. Experiments with crowdsourced re-annotation of a POS tagging data set. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (ACL)*, pages 377–382, Baltimore, Maryland, June. Association for Computational Linguistics.

Ziheng Huang, Jialu Zhong, and Rebecca J. Passonneau. 2015. Estimation of discourse segmentation labels from crowd data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2190–2200, Lisbon, Portugal, September. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–13, San Diego, CA, USA, May.

Sandra Kübler and Desislava Zhekova. 2011. Singletons and coreference resolution evaluation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011 (RANLP)*, pages 261–267, Hissar, Bulgaria. Association for Computational Linguistics.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 03.

Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. 2014. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment*, 8(4):425–436, December.

Maolin Li, Nhung Nguyen, and Sophia Ananiadou. 2017. Proactive learning for named entity recognition. In *BioNLP 2017*, pages 117–125, Vancouver, Canada, August. Association for Computational Linguistics.

Maolin Li, Arvid Fahlström Myrman, Tingting Mu, and Sophia Ananiadou. 2019. Modelling instance-level annotator reliability for natural language labelling tasks. 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) (NAACL)*, pages 2873–2883, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Bill Yuchen Lin, Dong-Ho Lee, Frank F. Xu, Ouyu Lan, and Xiang Ren. 2019. AlpacaTag: An active learning-based crowd annotation framework for sequence tagging. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL)*, pages 58–63, Florence, Italy, July. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, HLT '05, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9(November):2579–2605.

An Thanh Nguyen, Byron Wallace, Junyi Jessy Li, Ani Nenkova, and Matthew Lease. 2017. 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) (ACL)*, pages 299–309, Vancouver, Canada, July. Association for Computational Linguistics.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain Marshall, Ani Nenkova, and Byron Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pages 197–207, Melbourne, Australia, July. Association for Computational Linguistics.

Silviu Paun, Jon Chamberlain, Udo Kruschwitz, Juntao Yu, and Massimo Poesio. 2018. A probabilistic annotation model for crowdsourcing coreference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1926–1937, Brussels, Belgium. 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) (NAACL)*, pages 2227–2237. Association for Computational Linguistics.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pages 1–27. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea, July. Association for Computational Linguistics.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *Journal of Machine Learning Research (JMLR)*, 11(Apr):1297–1322, August.

Filipe Rodrigues and Francisco Pereira. 2018. Deep learning from crowds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, February.

Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. Sequence labeling with multiple annotators. *Machine Learning*, 95(2):165–181, May.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to wikipedia. *University of Massachusetts, Amherst, Technical Report UM-CS-2012*, 15.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 254–263, Honolulu, Hawaii, October. Association for Computational Linguistics.

Beth M. Sundheim. 1995. Overview of results of the MUC-6 evaluation. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Alena Tsvetkova. 2020. Anaphora resolution in chinese for analysis of medical q&a platforms. In *The 9th CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC)*, pages 490–497, Zhengzhou, China, October. Springer.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding (MUC)*, MUC6 '95, pages 45–52, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

YaoSheng Yang, Meishan Zhang, Wenliang Chen, Wei Zhang, Haofen Wang, and Min Zhang. 2018. Adversarial learning for chinese ner from crowd annotations. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 1627–1634, New Orleans, Louisiana, USA.

Li'ang Yin, Jianhua Han, Weinan Zhang, and Yong Yu. 2017. Aggregating crowd wisdoms with label-aware autoencoders. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1325–1331, Melbourne, Australia.

J. Zhang, V. S. Sheng, and J. Wu. 2019. Crowdsourced label aggregation using bilayer collaborative clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 30(10):3172–3185, October.

Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541–552, January.