Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Adaptive deep learning for entity disambiguation via knowledge-based risk analysis

Youcef Nafa, Qun Chen *, Boyi Hou, Zhanhuai Li

School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, China
Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Xi'an, Shaanxi, China

ARTICLE INFO

Keywords:
Risk analysis
Entity disambiguation
Adaptive training

ABSTRACT

The state-of-the-art performance on entity disambiguation has been reached by deep neural networks. However, the task remains very challenging due to the complexity of natural language. Moreover, the target data distribution is often different from that of training data. In this paper, we address the limitation of deep entity disambiguation from the perspective of misprediction risk. We propose a knowledge-based approach of risk analysis for entity disambiguation, and leverage it to enable adaptive deep learning. The proposed approach generates risk features by extracting evidences from the knowledge base, and then models them as a linearly-weighted random vector where an attention mechanism is used to focus on the most significant components. Finally, it estimates misprediction risk of the aggregated probability distribution via the Conditional Value-at-Risk metric. Furthermore, we demonstrate how to utilize risk analysis results in adaptive deep learning via two-phase training, the first phase fits on labeled training data while the second one minimizes misprediction risk on unlabeled target data. We evaluate the performance of the proposed approach on benchmark datasets through a comparative study. Our thorough experiments demonstrate that it can detect mispredictions more accurately than existing alternatives and can substantially improve the performance of deep learning models.

1. Introduction

The goal of Entity Disambiguation (ED) is to automatically link mentions of entities from a given document to their corresponding entities in a knowledge base. A mention is a word, a phrase or an expression that refers to an entity. An entity is something that exists as a subject or an object, physically or abstractly; such as a person, a country, a language or a chemical process to name a few. For instance, the sentence *"On 29 December 2022, Brazilian former footballer Pelé died aged 82 in Morumbi, São Paulo"* contains the mentions to the football player *Pelé* and the *Morumbi* destrict in *São Paulo, Brazil*. In practice, the search for the right entity for a given mention is limited to a set of candidates from the knowledge base. A candidate is an entity that has a high probability of being referenced by a given mention.

Entity disambiguation is usually preceded by Named Entity Recognition (NER), which highlights mentions within the text. The process composed of NER followed by ED is referred to as Entity Linking. It is considered to be a fundamental task in the natural language processing field (Kataria et al., 2011; Ratinov et al., 2011; Sen, 2012; Zheng et al., 2010). Its importance lies in its ability to endow free text on the web with encyclopedic knowledge which improves the experience of its readers. The ability to highlight entities such as persons, countries, or organizations on the web would perform a grounding of the relevant concepts via their knowledge base entries. Moreover, it represents a crucial step for knowledge base population (Ji & Grishman, 2011), as this task would otherwise require laborious efforts. And, it also eases the task of Question Answering (QA) when the entities mentioned in the question can be easily located inside documents on the web.

While recognizing mentions could be achieved by exploiting the sentence structure via methods such as parts-of-speech tagging and dependency tree parsing, ED can pose a bigger challenge due to the need for capturing the context of the mentions to disambiguate between extremely similar entity names. This requires at least some understanding of the relationship between the entity and common contexts in which it is often mentioned. For example, the name Eagles may refer to more than 40 entities in Wikipedia, like Colorado Eagles from the American Hockey League, Philadelphia Eagles in the National Football League (NFL) and Newcastle Eagles in the British Basketball League. In such case, the correct entity is found by considering the sport being

* Corresponding author at: School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, China.
E-mail addresses: youcef.nafa@mail.nwpu.edu.cn (Y. Nafa), chenbenben@nwpu.edu.cn (Q. Chen), ntoskrnl@mail.nwpu.edu.cn (B. Hou), lizhh@nwpu.edu.cn (Z. Li).
URL: https://chenbenben.org (Q. Chen).

https://doi.org/10.1016/j.eswa.2023.122342
Received 7 June 2023; Received in revised form 23 October 2023; Accepted 24 October 2023
Available online 28 October 2023

discussed, the league, the city name or even by the presence of another entity such as a player in the team or the team's president.

A lot of work has been done through the years to tackle the problem of ED. These range from machine learning supervised models (Bunescu & Paşca, 2006; Fleischman & Hovy, 2004; Milne & Witten, 2008), clustering models (Pedersen et al., 2005), ranking models (Dredze et al., 2010; Ratinov et al., 2011; Zheng et al., 2010) and probabilistic hierarchical solutions (Han & Sun, 2011, 2012; Kataria et al., 2011; Lazic et al., 2015). More recently, many Deep Learning models were proposed for ED (Ayoola et al., 2022; Cao et al., 2018; Fang et al., 2016; Ganea & Hofmann, 2017; Globerson et al., 2016; Huang et al., 2015; Le & Titov, 2018, 2019; Sevgili et al., 2020; Yamada et al., 2017, 2022; Yang et al., 2019). Most of these approaches rely on word or entity embeddings extracted from a large knowledge base. Neural Network models achieved state of the art results on all Entity Disambiguation benchmarks. Nonetheless, the task of ED remains very challenging due to the complexity of natural language on one hand, and the weak exposure to world knowledge from the models on the other hand. Its difficulty resides in solving the ambiguity by considering the whole sentence contents to decide which entity best matches the discussed topic and is the most coherent with the other mentioned entities in the entire mentioning document. Failing to do so may lead to mistakenly linking mentions to entities that, despite being textually similar, do not represent actual matches.

Aside from the inherent linguistic challenges, another issue worth noting is the distribution shift between training data and target data. Fitting a model to documents from a certain domain dealing with specific topics and entities does not guarantee a consistency in performance on other domains. The difference across sampled datasets, even from the same domain, could still pose challenges due to the shift in distribution between samples. The fact that many recent models rely on vector representations of entities embedded in the same space does not fully solve this generalization problem across domains. Most of the models' learned weights still overfit to the specificities of data seen during training. In the interest of mitigating these hurdles, we find ourselves in need of techniques to identify the situations where the deep learning model faces novel inputs on which its output is at high risk of being wrong. Then, such valuable information is used to update the output so as to *adapt* it to the target input. These two steps lead us to propose a knowledge-based approach of risk analysis for ED and leverage it to enable adaptive deep learning.

Risk analysis is the process by which the misprediction risk of the Neural Network model is estimated via a risk metric such as Conditional-Value-at-Risk (CVaR). It was proposed in Chen et al. (2020) and implemented through the LearnRisk model. In this work, we aim to leverage world knowledge that is external to both the DNN model and the training data, to provide a better estimation for misprediction risk. We extend risk analysis as formulated in Chen et al. (2020) by harnessing a knowledge base, such as Wikipedia, to extract better and more accurate risk features. The motivation being that, basing risk feature extraction only on the training data limits the abilities of the risk model in learning misprediction-inducing patterns. Because, after all, the DNN has already seen the training data. So, training-data-based risk features, despite offering a different view of the data than that of the DNN model, still share the same data source. The proposed approach generates risk features by extracting evidences from the knowledge base, and then models them as a linearly-weighted random vector where an attention mechanism is used to attend to the most significant components. Finally, it estimates misprediction risk by the aggregated probability distribution via the risk metric of Conditional Value-at-Risk (CVaR).

Risk analysis was also used to perform classifier adaptation in the cases of domain or distribution shifts (Chen et al., 2022). We demonstrate how to employ the results of risk analysis in adaptive deep learning via two-phase training, the first phase on labeled training data and the second one on unlabeled target data by minimizing misprediction risk. Improving misprediction risk estimation via external knowledge has the potential to improve the adaptation task.

The contributions of this work are summarized as follows

- We propose a novel knowledge-based approach of risk analysis for the task of Entity Disambiguation. As opposed to limiting the risk metrics to training data, the newly devised metrics make use of diverse sources, which include embeddings, knowledge base data and classifier representations, offering more evidence supporting misprediction risk estimation.
- We present an attention-based weighting mechanism to aggregate risk features' class membership probabilities into one distribution. The trainable attention module that we incorporated allows better modeling of the risk features' contributions to the final score relative to each candidate entity.
- We present an adaptive deep learning approach for the task of ED that can tune a deep model towards a particular workload by minimizing its misprediction risk. It is achieved by two-phase training, the first phase on labeled training data and the second one on unlabeled target data by minimizing misprediction risk.
- We perform an extensive empirical study to validate the efficacy of the proposed approaches. Our experiments on real benchmark data show that the proposed risk analysis approach can identify mispredictions with considerably higher accuracy than the existing alternatives; the proposed adaptive deep learning approach can also outperform the SOTA deep models by considerable margins.

The rest of this paper is organized as follows: Section 2 discusses the related work. In Section 3.1, we review the risk analysis framework. In Section 4, the approach of knowledge-based risk analysis is proposed and in Section 5 the adaptive training method is presented. Section 6 presents the empirical evaluation results. Finally, Section 7 concludes the paper.

2. Related work

Entity disambiguation (ED) is a fundamental task in the field of natural language processing (Kataria et al., 2011; Ratinov et al., 2011; Sen, 2012; Zheng et al., 2010), and an important step in knowledge base population (Ji & Grishman, 2011). Entity disambiguation aims to automatically link mentions of entities from a given document to their corresponding entities in a knowledge base. A lot of work has been done through the years to tackle the problem of ED (Alam et al., 2022). Earlier works focused on linking names from text to entities that represent people (Bagga & Baldwin, 1998; Fleischman & Hovy, 2004; Mann & Yarowsky, 2003; Pedersen et al., 2005). Some approaches attempted to add some background knowledge from social networks to capture name-relatedness (Bekkerman & McCallum, 2005; Malin et al., 2005; Minkov et al., 2006). Another line of approaches exploited Wikipedia as background knowledge for ED (Bunescu & Paşca, 2006; Cucerzan, 2007; Han & Zhao, 2009).

The proposed solutions for ED span across many paradigms. Some methods used machine learning supervised models. Milne and Witten (2008) proposed a machine learning solution that uses the links between Wikipedia articles for training a disambiguation model. The authors came up with commonness and relatedness features of entities and used them to transform the input to multiple machine learning algorithms, such as Naive Bayes, Support Vector Machine (SVM) and decision tree. Bunescu and Paşca (2006) trained a SVM Kernel to exploit the rich structure of the knowledge embedded in an online encyclopedia for entity disambiguation. The authors used Wikipedia's categories, disambiguation pages, redirect pages, and hyperlinks. Fleischman and Hovy (2004) trained a Maximum Entropy model to estimate the probability that two names refer to the same individual. Multiple feature types were used including web features, name features and estimated statistics to name a few.

Other works treated the disambiguation problem as a learn-to-rank problem instead of a classification problem. Zheng et al. (2010) used a set of similarity features, context features and miscellaneous features fed to a ranking Perceptron or a ListNet model for ranking. Dredze et al. (2010) used a ranking SVM, trained to rank the right entity higher than the other candidates. Similarly, Ratinov et al. (2011) used a two-step process consisting of a ranker model that obtains the best disambiguation for each mention, followed by a linker that chooses whether to link the mention to Wikipedia. Pedersen et al. (2005) presented an unsupervised approach to the disambiguation of names by clustering them into groups of entities according to statistically significant bigram features.

There have been many probabilistic solutions based on hierarchical models. Han and Sun (2011) proposed a generative probabilistic model that can make use of heterogeneous entity knowledge for the entity linking task. The knowledge consisted of popularity knowledge, name knowledge and context knowledge. Kataria et al. (2011) and Han and Sun (2012) modeled the problem using topic modeling. Plato (Lazic et al., 2015) presented a selective context probabilistic model for ED.

Neural network-based approaches have attained compelling results on this task (Cao et al., 2018; Fang et al., 2016; Ganea & Hofmann, 2017; Globerson et al., 2016; Huang et al., 2015; Le & Titov, 2018, 2019; Sevgili et al., 2020; Yamada et al., 2017, 2022; Yang et al., 2019). Most of these approaches rely on word or entity embeddings extracted from a large knowledge base. These embeddings can be fixed such as skip-gram embeddings (Mikolov et al., 2013) and node2vec (Grover & Leskovec, 2016) or contextualized such as ELMo (Peters et al., 2018) (e.g. used in Shahbazi et al. (2019)) and BERT (Devlin et al., 2019) (e.g. used in Broscheit (2019), Ling et al. (2020) and Yamada et al. (2022)). Many works make use of the attention mechanism (Bahdanau et al., 2015), be it to ensure entity coherence per document (Globerson et al., 2016), to focus only on important context words predictive of the mentioned entity (Ganea & Hofmann, 2017) or to incorporate entity type information (Nie et al., 2018). GENRE was proposed in Cao et al. (2021) and its multilingual version mGENRE in De Cao et al. (2022), a system that retrieves entities by generating their unique names using a transformer-based architecture. Kannan Ravi et al. (2021), Kolitsas et al. (2018) and Févry et al. (2020) attempted to solve end-to-end entity linking (Mention Detection followed by Entity Disambiguation) in a single one-fits-all solution. Ayoola et al. (2022) introduced an ED model that performs entity linking by reasoning over a symbolic knowledge base in a differentiable manner. In this paper we experiment with Dynamic Context Augmentation (DCA) (Yang et al., 2019). It is a global model that accumulates knowledge from previously linked entities as dynamic context to better inform later mention disambiguation decisions.

There have also been graph-based approaches (Al-Moslmi et al., 2020). Alhelbawy and Gaizauskas (2014) used Page-Rank on a weighted graph of candidates. Cao et al. (2018) used a Graph Convolutional Network to incorporate both local contextual features and global coherence information for entity linking. Sevgili et al. (2019) integrated information from a knowledge base with textual information using graph embeddings. Hu et al. (2020) introduced an end-to-end graph neural network model that utilizes semantic information for entity disambiguation.

Risk analysis, also referred to in the literature as confidence ranking or trust scoring. The goal of this task is to identify instances where a deployed model is succeptible of misprediction. Many solutions were proposed to tackle this problem, ranging from simple methods that make use of the model's outputs (Hendrycks & Gimpel, 2017) to more complex solutions that have other characteristics such as interpretability and learnability (Chen et al., 2020; Corbière et al., 2019; Jiang et al., 2018; Zhang et al., 2014). Of the latter type, LearnRisk (Chen et al., 2020) is an interpretable and learnable framework for Entity Resolution that builds a dynamic risk model tuned towards a given model and a target workload. In this framework, the risk is measured by the Value-at-Risk (VaR) metric from financial risk modeling (Tardivo, 2002). Other than misprediction detection, the concept of risk was also used in the area of machine learning for different ends such as recommendation risk (Xiao et al., 2020), fairness (Williamson & Menon, 2019) and distribution-free uncertainty estimation (Bates et al., 2021).

3. Background and challenges

Originally proposed in Chen et al. (2020) and substantiated in the LearnRisk model, the risk analysis framework consists of three steps: *Risk Feature Generation*, *Risk Model Construction* and *Risk Model Training*. We first define the problem of Risk Analysis for ED, and then, we summarize risk analysis steps and highlight some of their limitations.

3.1. Problem statement

Given a set of documents D where each document $a \in D$ consists of one or more phrases about a certain topic. These phrases will naturally refer to certain real-world entities such as: places, famous people, sports teams or scientific concepts to name a few. The part of the phrase which refers to an entity is called a *mention* of that entity. Mentions may be as simple as the entity name itself or a part of it, or they may be as complex as alternative appellations or abbreviations of entity names. A document a will consist of a set of entities M^a where each mention $m \in M^a$ refers to an entity e from the set of all entities E. The goal of Entity Disambiguation is to find all entities referred to by the mentions $m \in M^a$ for all the documents in D. So, a solution $\hat{S}$ for D will be in the form of pairs $(m_i, \hat{e}_i)$ where $\hat{e}_i$ is the predicted entity. The ground truth solution S for D is the set of pairs (m_i, e_i) where e_i is the correct entity mentioned by m_i. In practice, ED solutions are provided with a set of heuristically selected candidate entities C_i for each mention m_i to significantly reduce the search space from E to C_i.

The task of Risk Analysis is to rank the pairs $(m_i, \hat{e}_i)$ such that the wrong pairs are ranked higher than the correct pairs. Obviously, Risk Analysis cannot have access to the ground truth ED solution S. Instead, S is used to measure the performance of Risk Analysis in detecting mispredictions. The metric used to measure the performance of Risk Analysis has to take into account the whole ranking of pairs such that the ratio of detected mispredictions at every portion of the ranking is reflected. One such metric is the Receiver Operating Characteristic (ROC) curve (Fawcett, 2006) and its accompanying Area Under the ROC curve (AUROC) which were used in previous work (Hendrycks & Gimpel, 2017).

Let TP denote the number of true positives, FP the number of false positives, TN the number of true negatives and FN the number of false negatives. A positive is a mispredicted pair $(\hat{e}_i \neq e_i)$ while a negative is a correctly predicted pair $(\hat{e}_i = e_i)$. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at different thresholds such that $TPR = \frac{TP}{TP+FN}$ and $FPR = \frac{FP}{FP+TN}$. The ROC curve shows the trade-off between misprediction detection (true positives) and the incurred cost in correct predictions (false positives). So, to extract one score that reflects the quality of the solution, the area under the ROC curve can reflect the probability of assigning a higher score to a randomly chosen mispredicted pair than to a randomly chosen correctly predicted pair (Fawcett, 2006). Fig. 1 illustrates an example of two ROC curves for two methods: method A and method B. In this scenario, method A is better performing than method B while the diagonal (blue) line corresponds to a random performance.

3.2. Risk feature generation

Risk features are generated in the form of rules by using a one-sided decision tree algorithm. Each rule is an implication where the left-hand-side is a conjunction of conditions on scalar metrics and the right-hand-side is the class label. The algorithm ensures that the generated rule-set possesses two properties: discrimination between

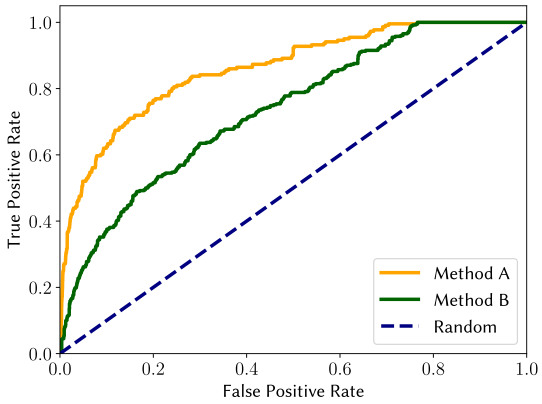


Fig. 1. Receiver Operating Characteristic curve.

classes and high-coverage of training data. In contrary to the common scenario where a rule is used to label examples with class labels, a risk feature concerns only a single class. As a result, risk features can be seen as indicators of the classifier's mispredictions relying on the knowledge encoded in the corresponding rules.

For a complex NLP problem such as ED, the metrics used for risk-rule generation cannot be extracted solely from training data since it does not provide enough knowledge to decide whether an entity matches a certain mention. More knowledge about the entity is needed to map it to the right mention and the right context. In addition, the data does not cover enough entities and mention scenarios which makes it hard to generalize risk analysis to novel inputs relying only on text-based rules extracted from it. As a result, the exploitation of a knowledge base as well as word and entity embeddings, as done in our approach, remedies the problem of labeled-data support.

Knowing that each single input in ED is an entity mention and a set of possible candidates, making it a multi-class problem, risk feature generation can still be performed on each mention-candidate combination individually.

3.3. Risk model construction

With the risk features generated from the previous step, it is now possible for the risk model to assess the classifier's predictions supporting its decision with interpretable explanations. Borrowing from investment theory, LearnRisk treats each instance's class membership probability distribution as a portfolio reward. It models the distribution as the aggregation of the class membership distributions of its features, treating them as stock rewards.

Formally, the class membership probability of a single datum (d_i, y_i), consisting of input d_i and label y_i, is modeled by a random variable π_i. π_i follows a normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$ with expectation μ_i and variance σ_i^2. $\mathcal{N}$ is truncated to the $[0,1]$ interval. For a set of n_F risk features $F = \{f_1, f_2, \dots, f_{n_F}\}$, let $\mathbf{w} = [w_1, w_2, \dots, w_{n_F}]$ denote their corresponding weights vector. Assume that $\boldsymbol{\mu}_F = [\mu_{f_1}, \mu_{f_2}, \dots, \mu_{f_{n_F}}]^T$ and $\boldsymbol{\sigma}_F^2 = [\sigma_{f_1}^2, \sigma_{f_2}^2, \dots, \sigma_{f_{n_F}}^2]^T$ represent the expectation and variance vectors corresponding to the random variables $\pi_{f_j} \sim \mathcal{N}(\mu_{f_j}, \sigma_{f_j}^2)$. Where π_{f_j} denotes the class membership random variable for the feature f_j. Then, π_i is estimated via

$$\pi_i = \sum_{j=1}^{n_F} (\mathbf{x_i})_j \, w_j \, \pi_{f_j} \, .$$

By using the normal distribution properties, π_i's distribution parameters are estimated as follows:

$$\mu_i = (\mathbf{x}_i \times \mathbf{w}) \cdot \boldsymbol{\mu}_F \, , \qquad \sigma_i^2 = (\mathbf{x}_i \times \mathbf{w})^2 \cdot \boldsymbol{\sigma}_F^2 \, .$$

Where $\mathbf{x}_i$ is a mask vector for active risk features for d_i.

Besides the risk features generated in the form of rules, LearnRisk includes another extra risk feature, which is the classifier's output probability. The classifier outputs a probability $\hat{\pi}_i = P_\Theta(y_i = 1 | d_i)$ for d_i belonging to the positive class where Θ are the classifier's learned parameters. Then, the range of probabilities $[0,1]$ is split into n_b bins in order to discretize $\hat{\pi}_i$, where each bin has its corresponding μ and σ^2. The resulting mask vector $\mathbf{x_i} \in \{0,1\}^{n_F}$ then equals 1 for the risk features (rules or bins) that fit the pair d_i, and equals 0 otherwise. Where $n_F = n_r + n_b$ represents the total number of rules n_r and classifier probabilities binned to n_b bins, the mask vector definition is then as follows

$$(\mathbf{x_i})_j = \begin{cases} 1 & \text{if } (j \le n_r) \wedge (r_j \text{ fits } d_i) \\ 1 & \text{if } (n_r < j \le n_r + n_b) \wedge (\hat{\pi}_i \in [\frac{j-n_r-1}{n_b}, \frac{j-n_r}{n_b}]) \\ 0 & \text{otherwise,} \end{cases} \qquad 1 \le j \le n_r + n_b \, .$$

Provided with the distribution π_i for d_i, its risk is measured by the metric of VaR (Tardivo, 2002).

The global weight vector $\mathbf{w}$ emphasizes risk features that are the most effective w.r.t the current classifier. However, for a problem such as ED, where the mispredictions can also be influenced by the current candidate entity, risk features are not only dependent on the classifier but also on the given candidate. Moreover, with more than a dozen features, the model ignores the potential dependence between risk features by treating their corresponding class membership random variables as independent. We propose to factor in entity information to learn $\mathbf{w}$ and use a single multivariate distribution to model risk feature random variables as a random vector. Note that at this step, other more robust risk measures can be applied. Subsequently, we consider the more coherent Conditional Value-at-Risk (CVaR) metric (Artzner et al., 1999).

3.4. Risk model training

In this last step, the risk model is fitted on the validation data by optimizing a learn-to-rank objective (Burges et al., 2005). The parameters tuned during the training process are: the risk feature weight vector $(\mathbf{w})$ and the variances $(\boldsymbol{\sigma}_F^2)$. Since the expectations $(\boldsymbol{\mu}_F)$ are calculated from the training data, they are treated as prior knowledge.

While we can consider all mention-candidate combinations for ED, from a risk analysis perspective, we are more concerned about the risk of the candidate predicted by the classifier. So, an aggregation of the individual risk scores is required to generate one risk score per mention. The reformulation of the ranking objective in this case is required (more details in Section 4).

Once trained, the risk model is applied to unseen data that is labeled by the same classifier. It is used to assess the misclassification risk and output risk scores for each prediction.

4. Knowledge-based risk analysis

In this section, we describe the approach of knowledge-based risk analysis. Let us assume the knowledge-base consists of entities and their textual descriptions, and that each text contains mentions of other entities in the same knowledge base. A mention is a word, a phrase or an expression that refers to an entity. For example, *China* and *PRC* are mentions for the entity *People's Republic of China*. The approach follows the risk analysis pipeline (Section 3.1). In what follows, the distinctive characteristics that are either added to or adapted from the original approach are explained in detail. These stem from the exclusive use of the knowledge-base and the multi-class nature of ED or represent desirable improvements. They can be categorized into five major procedures. Firstly, *Knowledge-base Evidence Extraction* is responsible for generating evidence for each known entity in the knowledge base. Then, *Risk Metrics Synthesis* generates similarity metrics that make use of the evidence from the previous step as well as word embeddings and textual data. After *Risk Feature Generation* using the synthesized metrics,

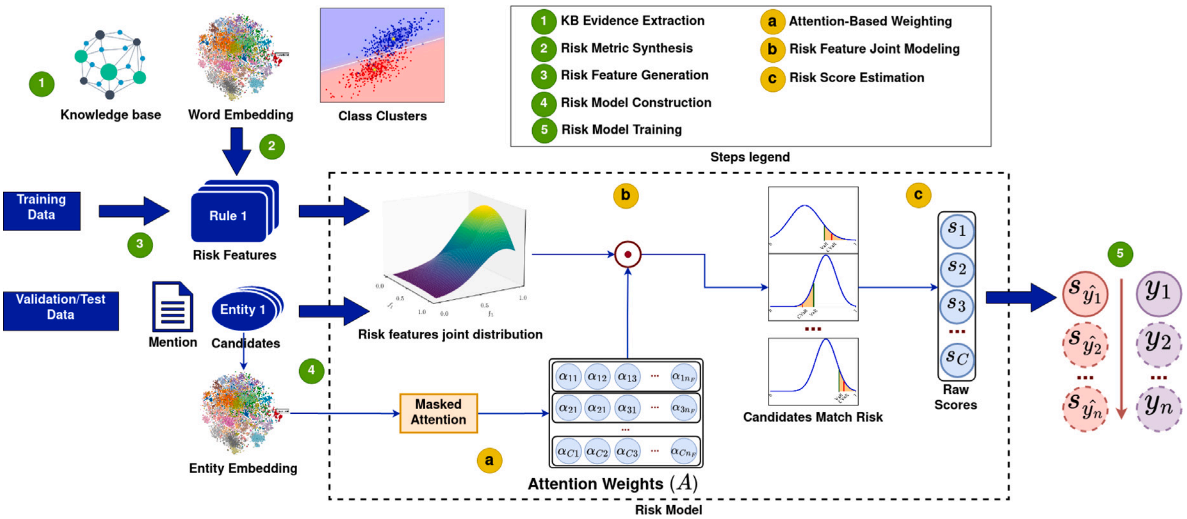


Fig. 2. Knowledge-based risk analysis framework.

Risk Model Construction describes the risk model structure and motivates its design choices. Finally, *Risk Model Training* outlines the training objective and the estimation method for the multi-class scenario (as showcased in ED).

Fig. 2 shows the knowledge-based risk analysis framework. The aforementioned steps of the risk analysis process are numbered chronologically from 1 to 5.

4.1. Knowledge-base evidence extraction

Evidence is defined as the individual concepts that co-occur with an entity's mentions, i.e. words, entities or notions that appear the most in the context of the mentions relating to the entity. The task, is to first transform the raw linked text data into a common numerical representation. Then, three types of entity-level evidences are extracted, namely, Topics, Keywords and Cross-links.

4.1.1. Common representation

Let E be the set of all entities in the knowledge base. For every entity $e \in E$ we can extract n_e mentions of a given window size δ. The set of mentions of e, M^e, is defined as:

$$M^e = \{M_1^e, \dots, M_{n_e}^e\}.$$

Let M be the set of all mentions, i.e. $M = \cup_{e\in E} M^e$. We create a vocabulary V of n_V tokens excluding stop-words, top frequent words and rare words. V is extracted from the set of all mentions M. Then, we limit every mention M_i^e to the words in V resulting in $\hat{M}_i^e$. Let I_i^e represent the indices of the words in $\hat{M}_i^e$. $\hat{M}_i^e$ is transformed into one-hot representation $R_i^e \in \mathbb{R}^{n_V}$ such that:

$$R_{ij}^e = \begin{cases} 1 & \text{if } j \in I_i^e \\ 0 & \text{otherwise} \end{cases}, \quad 1 \le j \le n_V. \tag{1}$$

4.1.2. Topics

The first type of evidence that can be held against classifier predictions, is whether the predicted entity for a given mention and context differ in their topics. As such, the topics act as *coarse-grained* evidence. We train a topic model on the set of word counts per entity using LDA (Latent Dirichlet Allocation (Blei et al., 2001)) with n_T topics. LDA model is trained on $\cup_{e\in E} R^e$ where $R^e = \sum_{i=1}^{n_e} R_i^e$, resulting in the function lda $:\mathbb{N}^{n_V} \rightarrow \mathbb{R}^{n_T}$ where the output is a probability distribution over n_T topics (refer to Section 6 for details on how to choose n_T).

Then, we transform each entity's mentions list using the lda function ($\theta_e = \text{lda}(R^e)$, $\theta_e \in \mathbb{R}^{n_T}$) and keep the top-k topics ranked by θ_e as the topic list of e, namely T_e.

4.1.3. Keywords

As opposed to topics, this type of evidence is *fine-grained*. Keyword evidence is represented as the individual token influence on the relationship between the mention's context and the entity. We apply the TF-IDF algorithm on $\cup_{e\in E} R^e$, i.e. treating each entity's mentions as a single document (sum of the one-hot vectors). Then, we get the TF-IDF scores of words for each entity.

Specifically, for every word $w_i \in V$, we calculate its term-frequency in the mentions of e (i.e. $\text{TF}(w_i, M^e)$), and its document-frequency ($\text{DF}(w_i)$) as the number of entities containing w_i in any of their mentions (i.e. $\text{DF}(w_i) = |\{M^e : w_i \in M^e, e \in E\}|$). Then, the TF-IDF score $\text{TF-IDF}(w_i, e)$ is calculated as

$$\text{TF-IDF}(w_i, e) = \text{TF}(w_i, M^e)\,\text{IDF}(w_i),$$

where $\text{IDF}(w_i) = \ln(\frac{|E|}{\text{DF}(w_i)})$

Finally, the Keyword evidence of an entity e is given by the set $K_e = \{w_i \in V, \text{TF-IDF}(w_i, e) \ge \gamma\}$ where γ is a threshold.

4.1.4. Cross-links

The previous types of evidences focus on recognizing entities from their context, which means they are local evidences. It is equally important to model links between entities in order to learn which ones are semantically close to each other and which are not. In this scenario, the goal is not to find entities that are synonymous. Instead, the tendency of entities to co-occur in the same document is what we are interested in. It is worthy of noting that even topic information is not enough to cover such links. For instance, a mention of an athlete's hometown in a document about sports does not necessarily mean the town itself is related to sports, yet it tends to co-occur with the athlete's mentions.

Concretely, the evidence is pretty straightforward to extract and represent. The cross-link data extraction step can be performed while extracting the mentions, because as assumed in the knowledge base, every mention actually resides in a document describing an entity. So, the set of mentions of a target entity e comes also with a set of source entities I_e which are considered e's in-links. Similarly, the set of e's out-links is essentially the set of entities O_e referred to by the mentions contained in e's document in the knowledge base.

Finally, the list of e's cross-links L_e, consisting of the entities either referring to e (in-links) or referred to by e (out-links) is represented as the set $L_e = I_e \cup O_e$. L_e can also be seen as the set of neighbors of e in a knowledge-graph where nodes represent entities and edges represent mentions.

4.2. Risk metrics synthesis

As seen in Section 3.2, risk rules operate on conditions and thresholds involving scalar metrics. For accurate risk analysis, high quality metrics need to be devised to ensure the generation of discriminative and high-support rules. At this step, metrics are generated from the dataset using three diverse sources: the knowledge base, word embeddings and classifier representations. The input dataset, for which metrics are to be computed, is transformed from a set of mention-candidates pairs $(m_i, \{e_{ij} : j \in [C]\})$ where C is the number of candidates, to an ordered list of mention-candidate pairs $[m_i, e_{ij}]$. Each mention m_i is surrounded by a left context c_i^l and a right context c_i^r making up the full mention context $c_i = c_i^l + m_i + c_i^r$. Wherever the i and j subscripts are omitted for readability, m, c and e refer to m_i, c_i and e_{ij} respectively.

The metrics are categorized according to their sources as detailed below.

4.2.1. Knowledge base

- **Topic coverage score** measures the coverage of the topics in the mention context c by the topics of the candidate entity e as:

  $$\text{topic\_sim} = \frac{|T_c \cap T_e|}{|T_c|},$$

  where T_c contains the top-k topics according to $\theta_c = \text{lda}(c)$ making up the set of topics of c, and T_e is the topic list of e from Section 4.1.2.
- **Different top-topic** is a binary score that returns 1 when the top topic –according to the LDA score– for the mention context and the entity are different, otherwise it returns 0.

  $$\text{diff\_top\_topic} = \begin{cases} 1, & \text{if } argmax(\theta_c) \neq argmax(\theta_e) \\ 0, & \text{otherwise} \end{cases}.$$

- **Context similarity score** is a ratio of the common tokens between the context c and the entity e given by the Jaccard index:

  $$\text{context\_text\_sim} = \frac{|K_c \cap K_e|}{|K_c \cup K_e|},$$

  where K_c is the set of tokens in $c \cap V$ and K_e the set of keywords of e from Section 4.1.3.
- **Document coherence score** is the number of document entities in which m occurred, that are also cross-links of e.

  $$\text{coherence} = |E_d \cap L_e|,$$

  where E_d is the set of ground-truth entities linked to mentions of document d and L_e is the list of e's cross-links from Section 4.1.4. At test time, E_d is the set of entities predicted by the classifier, as there are no ground-truth labels.
- **Mention-candidate prior** $\hat{p}(e|m)$ is estimated from the knowledge base's #(m,e) counts. It is often used to pre-rank candidates and limit their number to C candidates prior to classifier training (Ganea & Hofmann, 2017).

4.2.2. Word embeddings

Let $B^w \in \mathbb{R}^{n_V \times d_w}$ be a word embedding matrix of outer dimension d_w and $emb(S)$ (defined below) a function that returns the average embedding of a sequence of tokens S filtered by the vocabulary V. Averaging tokens is a simple yet effective baseline for sentence representation (Kenter et al., 2016).

$$emb(S) = \frac{1}{|S \cap V|} \sum_{w \in (S \cap V) \wedge V_k = w} B_k^w.$$

Four total word embedding-based metrics are extracted by considering the mention and the context and using two variants for the similarity function ($sim()$ below) in each case. Cosine and Euclidean similarity functions are considered.

- **Mention embedding similarity** measures the similarity between the mention phrase m and the candidate entity's name e by transforming them into dense representations. Then the similarity is measured via

  $$\text{mention\_emb\_sim} = sim(emb(m), emb(e)).$$

- **Context embedding similarity** also measures the similarity between averaged token vectors. Only that the left-hand side is actually the average over the whole context c.

  $$\text{context\_emb\_sim} = sim(emb(c), emb(e)).$$

4.2.3. Classifier representations

- **Positive class distance** (resp. **Negative class distance**) is simply the distance between the classifier's representation of (m, e) and the cluster center of all matching (resp. non-matching) mention–entity pairs from labeled training data D^T. The clusters for the positive and negative classes are $\{(m_i, e_{ij}) \in D^T : y_i = j\}$ and $\{(m_i, e_{ij}) \in D^T : y_i \neq j\}$ respectively.

4.3. Risk model construction

In this subsection, the steps required for risk model construction are outlined. The three steps are highlighted in Fig. 2 using letters **a**, **b** and **c**; corresponding to Attention-Based Weighting, Risk Features Joint Modeling and Risk Score Estimation, respectively.

4.3.1. Attention-based weighting

As seen in Section 3.1, the ultimate goal of the risk model is to learn risk feature weights for a given classifier so that it can model the class membership distribution for each input instance. For ED, each input $d_i = (m_i, \{e_{ij} : j \in [C]\})$ consists of a mention m_i with up to C candidates $\{e_{ij} : j \in [C]\}$. This means that in order to apply the risk model to d_i, each candidate e_{ij} has to be considered separately, where the label is *matching* for the correct candidate and *non-matching* for the remaining candidates. Recalling that risk features are represented by rules that operate on a set of metrics, it is quite limiting to learn one single risk feature weighting for all potential inputs with a large set of entities. It is possible to fall into situations where a given risk feature is highly predictive of the misclassification risk on a candidate entity, while the same feature is not indicative of the risk on another candidate. For example, an entity that is easily disambiguated by its top topic would have a higher weight for the risk features that operate on topic-based metrics (such as diff_top_topic), while an entity that is commonly mentioned in passages from various topics would have its risk estimated better via risk features from another category. So, the risk model not only learns which risk features are more predictive of mispredictions from the classifier, it would further learn the ones that work best for the candidate entity being predicted.

To achieve this functionality, risk feature weights need to be computed via a transformation that factors in the input candidate entity. We take inspiration from the *Dot-Product Attention* (Vaswani et al., 2017) to model the entity-aware risk-feature weights. This attention mechanism fits the desideratum we expressed as long as entities are represented by a dense embedding. Indeed, entity embeddings have been extracted and commonly used in machine learning solutions for ED (Fang et al., 2016). Let $Q^i \in \mathbb{R}^{C \times d_e}$ of dimension d_e denote the candidate entities' embeddings for d_i, $K \in \mathbb{R}^{d_e \times n_F}$ a feature mapping matrix and $X^i \in \{0,1\}^{C \times n_F}$ a Boolean mask (stacked $\mathbf{x}_i$ from Section 3.3 for C candidates). The candidate-level risk feature weights are stored in the rows of the matrix $A^i \in \mathbb{R}^{C \times n_F}$, which is computed following Eq. (2). The masked Softmax function ensures that the output weights sum to 1 and keeps only the risk features that fit to each mention-candidate pair using the Boolean mask. For compatibility, we keep the same terminology for the queries Q, keys K and attention weights A as in the original paper (Vaswani et al., 2017). So far, Q^i and X^i are the required inputs while K is learned during training.

$$A^i = masked\_softmax(K \cdot Q^i, X^i). \tag{2}$$

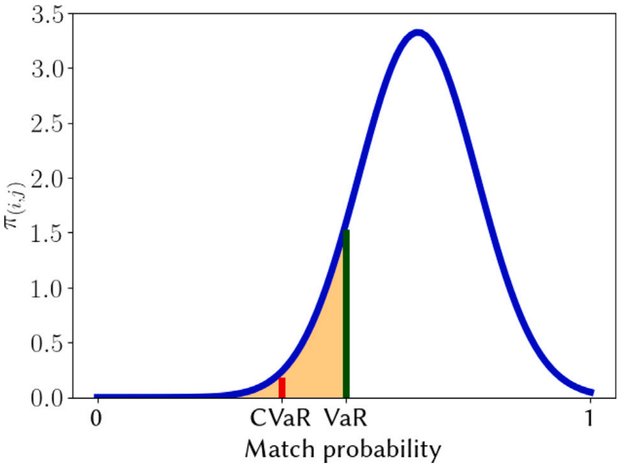


Fig. 3. Value-at-Risk versus Conditional Value-at-Risk.

4.3.2. *Risk features joint modeling*

The next step in the risk analysis framework is the estimation of the matching class membership random variable $\pi_{(i,j)}$ for each mention-candidate pair (m_i, e_{ij}). Contrary to the original risk model, which treats individual risk features as independent normally distributed random variables π_{f_j} (Section 3.3), we instead model them as components of a multivariate normally distributed random vector $\Pi_F \sim \mathcal{N}_{n_F}(\mu_F, \Sigma)$ where μ_F is its mean vector and Σ is a positive-definite matrix representing its covariance. μ_F consists of risk feature means that were estimated from the training data in the Risk-Feature Generation step, while Σ is learned during training. We argue that this modeling choice follows naturally from the fact that it is safer to assume risk features are dependent than independent. Indeed, risk features employing different similarity metrics are highly likely to express similar predictive ability in many situations. By considering the dependence characteristic, the linear combination of the marginal random variables would result in a more accurate joint distribution.

In, this scenario, $\pi_{(i,j)}$ will be the result of the linear combination of the weights A^i_j with the random vector Π_F. This design relies on two properties of the Multivariate Normal Distribution which we list below keeping the same notation:

1. The distribution of $\pi_{(i,j)} = A^i_j \cdot \Pi_F$, is univariate Normal with $\pi_{(i,j)} \sim \mathcal{N}(A^i_j \cdot \boldsymbol{\mu},\, {A^i_j}^T \cdot \Sigma \cdot A^i_j)$.
2. Any subset of the random vector Π_F has a marginal distribution that is also multivariate normal.

As a result, only one random vector needs parameter estimation throughout the training process. The only change per input instance is the masked attention weight, which relies on the first property to reduce the distribution into univariate, and relies on the second property to be able to mask certain variates with each input. Notice how by being positive-definite, Σ ensures that the variance of the dot product remains positive. To maintain this property, Σ is actually trained via its Cholesky decomposition $\Sigma = U^T \cdot U$ where U is an upper triangular matrix with strictly positive diagonals. Σ is trained gradually by receiving updates for different combinations of variates at each iteration. Attention weights will naturally preserve the mean of the distribution to be in the probability range of $[0,1]$ because the weights themselves sum to 1.

4.3.3. *Risk score estimation*

Having estimated its parameters, $\pi_{(i,j)}$ now reflects the odds of e_{ij} matching with m_i. This is the step where risk estimation is performed. The original model used VaR (Section 3.1) as the base risk metric. This metric has been criticized in the financial risk modeling domain for various reasons. Artzner et al. (1997) and Artzner et al. (1999) noted that it has undesirable mathematical characteristics such as lack of subadditivity and convexity, and that it is not informative about the magnitude of the losses larger than the VaR level. These misgivings are absent in the Conditional Value-at-Risk (Artzner et al., 1999) measure, also referred to as Expected shortfall or Average Value-at-Risk, which represents a more coherent measure with better properties. In fact, CVaR is computed for a random variable Z in terms of VaR through the expression

$$\mathrm{CVaR}_\epsilon(Z) = \mathbb{E}[Z \,|\, Z > \mathrm{VaR}_\epsilon(Z)],$$

where $\mathbb{E}$ is the statistical expectation and ϵ is the confidence. Fig. 3 shows the VaR and CVaR values for the Normally distributed random variable $\pi_{(i,j)}$ with a confidence of 0.9.

CVaR is still applied in our model in the same way VaR was applied in LearnRisk, that is, it is measured on the truncated version of $\pi_{(i,j)}$ taking into account the current classifier's predicted label $\hat{y} \in [C]$. So, the raw risk scores for $(d_i, \hat{y}^i)$ are represented by the vector $\mathbf{s^i} \in [0,1]^C$ where C is the number of candidates and each $s^i_j \in \mathbf{s^i}$ is calculated as follows:

$$s^i_j = \begin{cases} \mathbb{E}[\pi_{(i,j)} \,|\, \mathrm{VaR}_\epsilon(\pi_{(i,j)}) < \pi_{(i,j)} < 1] & \text{if } j \neq \hat{y}^i \\ 1 - \mathbb{E}[\pi_{(i,j)} \,|\, 0 < \pi_{(i,j)} < \mathrm{VaR}_{1-\epsilon}(\pi_{(i,j)})] & \text{if } j = \hat{y}^i \end{cases}, \qquad (3)$$

noting that 0 and 1 are the truncation bounds. Eq. (3) deals with two cases: one where the candidate is predicted by the classifier as matching and one where it is not. Each case determines the side from which CVaR is estimated. Step (c) in Fig. 2 shows CVaR estimation on different candidates where on one candidate (the one predicted as matching) it is measured from the left while on the other ones it is measured from the right.

Then, substituting the truncated normal expectation definition in (3) and replacing $\mathrm{VaR}_\epsilon(\pi_{(i,j)})$ with the truncated normal quantile function $F^{-1}(\epsilon)$ of $\pi_{(i,j)}$, we get a closed form expression (Rockafellar & Uryasev, 2000)

$$s^i_j = \begin{cases} \mu_i - \sigma_i \frac{\phi(1) - \phi(F^{-1}(\epsilon))}{\Phi(1) - \Phi(F^{-1}(\epsilon))} & \text{if } j \neq \hat{y}^i \\ 1 - \mu_i + \sigma_i \frac{\phi(F^{-1}(1-\epsilon)) - \phi(0)}{\Phi(F^{-1}(1-\epsilon)) - \Phi(0)} & \text{if } j = \hat{y}^i \end{cases}, \qquad (4)$$

where Φ is the standard normal cumulative distribution function and ϕ is the standard normal probability density function.

The scores in $\mathbf{s^i}$ represent individual probabilities reflecting the misprediction risk per candidate entity. The final step is to get one score ρ_i given the classifier's predicted candidate $\hat{y}^i$. This can be simply achieved by returning the $\hat{y}^i$-th score $s^i_{\hat{y}^i}$

$$\rho_i = s^i_{\hat{y}^i}.$$

4.4. *Risk model training*

For training, a pairwise learn-to-rank objective (Burges et al., 2005) is optimized to rank the mispredicted examples higher than the correctly predicted ones. The objective first transforms the score difference of a pair (d^l, d^n) into a probability P_{ln} that d^l is ranked higher than d^n using a Sigmoid function.

$$P_{ln} = \frac{1}{1 + e^{-\sigma(s^l - s^n)}}, \qquad (5)$$

where s^l and s^n are the scores for d^l and d^n, respectively, and σ determines the shape of the Sigmoid function. Then, cross entropy loss (Eq. (6)) is used to penalize the deviation of the output probabilities from the desired rank labels $S_{ln} = \frac{1 + y_l - y_n}{2}$ where $y_l, y_n \in \{0,1\}$ are the misprediction labels (1 for mispredicted instances and 0 otherwise). In this manner, $S_{ln} \in \{-1, 1, 0\}$ indicates whether d^l should be ranked lower, higher or either way relative to d^n.

$$\Psi = -S_{ln}\, log P_{ln} - (1 - S_{ln})\, log(1 - P_{ln}). \qquad (6)$$

If we limit the pairs to those where the first instance should be ranked higher than the second instance, i.e. the first instance is mispredicted by the classifier while the second one is correctly predicted, The cost function can be simplified into Eq. (7) by substituting P_{ln} from Eq. (5) and setting $y_l = 1, y_n = 0$.

$$\Psi = log(1 + e^{-\sigma(s^l - s^n)})\,. \tag{7}$$

The ranking can be performed with regard to individual mention-candidate scores s^i_j (*pair loss* in Eq. (8) below) or with regard to the final mention-level scores ρ_i (*aggregate loss* in Eq. (9) below). Although, the most compatible objective with ED's multi-class nature is the latter formulation, the former may provide extra inductive bias by imposing the ranking on more mention-candidate combinations. Later in Section 6, we consider both options and evaluate their influence on training and performance.

$$\Psi_s = \sum_{l\in\{y^i:\hat{y}^i=y^i\}} \sum_{n\in\{y^k,\hat{y}^k:\hat{y}^k\neq y^k\}} \log\left(1 + e^{\sigma_s(\bar{s}^i_l - \bar{s}^k_n)}\right). \tag{8}$$

$$\Psi_\rho = \sum_{\hat{y}^i=y^i} \sum_{\hat{y}^k\neq y^k} \log\left(1 + e^{\sigma_\rho(\rho_i - \rho_k)}\right). \tag{9}$$

The parameters that are fitted during training are the attention mapping matrix K, the covariance matrix Σ and the learn-to-rank objective's Sigmoid shape parameter σ_s or σ_ρ. The loss function along with all the operations performed by the risk model to arrive at the final risk scores are differentiable. Which allows the application of gradient decent to optimize the objective.

5. Risk-based adaptation

Chen et al. (2022) showed that risk analysis can be of great value in the task of distribution shift mitigation. It was shown that minimizing the risk on unlabeled target data helps in fine-tuning the already-trained model and adapting it to the intricacies of the new data. This was further validated in the case of transfer learning, i.e. when the target data comes from an entirely different domain. In this section, we perform adaptive training on ED using the risk model proposed in this work. It is safe to expect that leveraging outside knowledge should have a positive impact on the adaptation task.

Risk-based adaptive training consists of two phases, a traditional training phase followed by a risk-based training phase. The first phase is the conventional fitting on the training set using a loss function L_t such as Mean Squared Error, Cross Entropy or Max-Margin loss between the predictions and target labels. The second phase, uses a risk model trained on the validation data to further fine-tune the classifier by minimizing misprediction risk on the target test data.

It can be argued that finetuning the model using risk minimization alone can cause catastrophic forgetting. This is a side effect of finetuning all the learnable parameters within the DNN model on new data, without access to the original training data. The solution often used to circumvent this phenomenon, is to finetune the model for a few epochs by using an extremely low learning rate coupled with an optimization schedule that potentially starts with a warm-up stage followed by a constant or decayed learning rate stage. The setting used for risk-based training and the hyperparameters are given in Section 6.3. Let $P_\Theta(e_{ij}|m_i)$ be the scoring function for a given mention m_i and a candidate entity e_{ij} using Θ the fitted weights during the first phase. The loss function for risk-based training is given in Eq. (10)

$$L_r = \frac{1}{n}\sum_{i=1}^{n} -\mathbb{1}_{[q,1]}(\rho_i)\; log(1 - max_j\, P_\Theta(e_{ij}|m_i))\,. \tag{10}$$

where n is the total number of mentions in the target test set, $\mathbb{1}_{[q,1]}(x)$ is an indicator function that returns 1 when $x > q$. q is the $\frac{1}{1-\tau}$-quantile forming a threshold on risk scores that limits the correction to the extremely high risk examples. In our experiments, τ was set to 0.95 meaning that q is the upper *ventile* (20-quantile).

6. Experiments

In order to evaluate the performance of the proposed approach, We conduct two types of experiments. In the first experiment we perform a comparison of misprediction detection methods. The second experiment consists in performing domain adaptation for a trained DNN model using risk analysis. We further perform an ablation study and qualitative analysis to deeply inspect the proposed model.

The DNN model chosen for the ED task is DCA (Dynamic Context Augmentation) introduced in Yang et al. (2019). It relies on word and entity embeddings to extract local features such as Mention–entity Prior, Context Similarity and Type Similarity (Ganea & Hofmann, 2017). In addition, it augments them with coherence features that act as a Dynamic Context Augmentation which takes the global context of the mention into account (the previously matched entities so far in the same document). The features are then transformed via a feed-forward 2-layer network to produce a score for each mention–entity pair. In the supervised setting, the model is trained to minimize a max-margin loss resulting in a ranking model for entities. DCA is a state-of-the-art (SOTA) model which achieves an F1-score of 94.64 on AIDA-CoNLL's test dataset (refer to Section 6.1). Moreover, as opposed to many recent SOTA models, its results are reproducible.[1]

6.1. Datasets

We validate our approach on six popular ED datasets:

AIDA-CoNLL The dataset from Hoffart et al. (2011) contains a training set of 946 documents, a validation set of 216 documents, and a test set of 231 documents.

WIKI and **CLUEWEB** datasets are automatically extracted from the Wikipedia and ClueWeb corpora, respectively (Guo & Barbosa, 2018). Each dataset is split into a training, validation and test sets with ratios 60%, 20% and 20% respectively.

MSNBC (Cucerzan, 2007), **AQUAINT** (Milne & Witten, 2008) and **ACE2004** (Ratinov et al., 2011) are individual testsets used for out-of-distribution experiments.

6.2. Risk analysis

Our comparative experiment consists of the following methods:

Baseline (Hendrycks & Gimpel, 2017) a baseline uncertainty technique that simply measures confidence based on the classifier's output probability. The predictions close to 1 are considered confident while the ones that are close to 0 are deemed uncertain. Hence, the score is measured by the complement of the probability $P_\Theta(e_{ij}|m_i)$ to reflect uncertainty rather than confidence $(1 - P_\Theta(e_{ij}|m_i))$.

TrustScore (Jiang et al., 2018) calculates a trust score based on the agreement between the classifier and a modified nearest-neighbor classifier. The data representation used for distance measurement is extracted from DCA's feature layer. We use the official implementation with default parameters.[2]

ConfidNet (Corbière et al., 2019) adds a confidence scoring neural module to the main network. We use a fully connected network module with 5 layers and ReLu activations to compute the confidence score, then, train the module using the True Class Probability (TCP) criterion.[3]

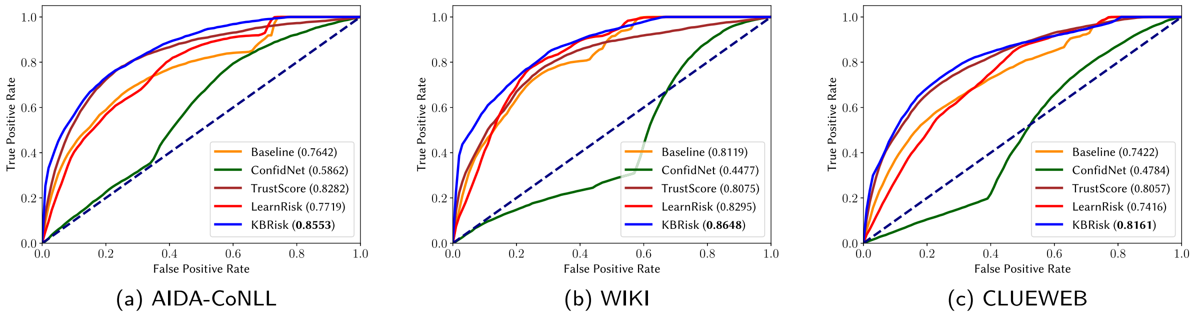
LearnRisk (Chen et al., 2020) uses a Risk Model as described in Section 3.1. It uses the similarity and difference metrics devised in the original paper.

KBRisk represents our approach.

[1] https://github.com/YoungXiyuan/DCA.

[2] https://github.com/google/TrustScore/.

[3] https://github.com/valeoai/ConfidNet.



(a) AIDA-CoNLL (b) WIKI (c) CLUEWEB

Fig. 4. Risk analysis comparative results. Figures (a)–(c) show the Receiver Operating Characteristic (ROC) curves on AIDA-CoNLL, WIKI and CLUEWEB respectively. Scores represent the area under the ROC curve.

Raw Wikipedia data is extracted from the official dumps[4] using wikiextractor.[5] For experimental purposes, the number of entities extracted was limited to cover the datasets in Section 6.1. Considering all $\sim 6M$ entities is possible if need be, given that hardware requirements and runtime are taken into account (by subsampling data or running on a cluster for example). All the available mentions for each entity were extracted by following hyperlinks resulting $\sim 13M$ mentions of a window size $\delta = 256$ tokens. The average number of entities per mention is 373.46. Scikit-learn was used to build a vocabulary V of $n_V = 20$ k tokens, train the LDA model and perform TF-IDF transformation. The threshold for entity keyword evidence γ is set to 0.1 (Section 4.1.3). The number of top k topics per entity is 10 (Section 4.1.2). Each dataset D is vectorized and transformed using TF-IDF and LDA models following the methodology presented in Section 4.1.

DCA model is trained with the default parameters in the supervised learning mode. The classifier's predictions $\hat{y}^i$ on the validation and test sets are then used to train and validate the risk model. The latter is trained for 100 epochs using Adam Optimizer with learning rate equal to 10^{-1} and a batch size of a 1500 mentions. The high learning rate is compensated by the large batch size (Smith et al., 2018) which accelerates training significantly. We made sure the loss kept decreasing throughout training with this configuration. CVaR confidence was set to 0.9 following Chen et al. (2020). The experiment was replicated *twenty* times with different random seeds (for K and Σ). The comparative results are given in Fig. 4. The Receiver operating characteristic (ROC) metric is used to compare misprediction detection performance (refer to Section 3.1). Methods with high true positive rate to false positive rate ratio are better at detecting mispredictions with lower budget. The plots in Fig. 4 show the averaged ROC graphs. The area under each ROC graph (AUROC) is noted in the corresponding legend. It is clear that the baseline shows competitive performance with an area under the curve way higher than the 0.5 bound (the diagonal line in the figure). ConfidNet is consistently under-performing despite being more expensive due to confidence module fitting. It even drops under the random guess line in some occasions. On the other hand, TrustScore shows good misprediction detection ability on all three datasets. LearnRisk attains good performance on WIKI dataset, and outperforms the baseline on all datasets. Finally, KBRisk outperforms all methods across all datasets. This is manifested in the larger AUROC compared with alternatives, with an increase of up to 4% in AUROC across datasets. Moreover, the ROC curve of KBRisk is consistently superior to other methods. Especially, in the low FPR side of the graph. Table 1 shows the p-values for the pairwise sample t-test between KBRisk and every other method. It can be seen that on almost all datasets, the AUROC scores seen in Fig. 4 are statistically significant with a confidence of 0.05. The only exception being the difference between KBRisk and TrustScore on the CLUEWEB dataset.

Table 1
Pairwise sample T-Test significance analysis. Cell values represent p-values between KBRisk and each alternative method for 20 runs across all datasets.

Method	AIDA-CoNLL	WIKI	CLUEWEB
vs Baseline	2.6307e−22*	1.8071e−22*	1.1176e−15*
vs ConfidNet	1.9637e−12*	3.6844e−41*	1.1114e−17*
vs TrustScore	8.7556e−10*	3.6436e−23*	0.054804
vs LearnRisk	2.802e−19*	2.296e−14*	3.8756e−15*

* p-values less than the confidence 0.05.

Indeed, by leveraging external knowledge, KBRisk was able to better detect the DNN model's mispredictions. In addition, fitting the risk model on validation mispredictions allowed for a better exploitation of the knowledge and a more precise assessment tailored to the DNN model.

6.3. Adaptation

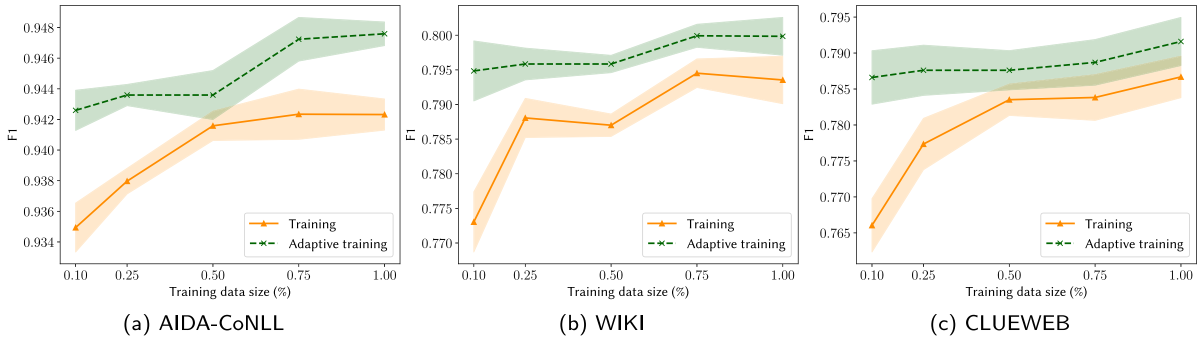
We perform adaptation experiments, following the approach in Section 5, on the three datasets presented above and evaluate the model's F1 score improvement, before and after adaptive training. The trained DNN model is further fitted on the loss L_r for 10 epochs with a learning rate of 10^{-5}. The classification layer is optimized with the unscaled learning rate, the inner layers use a smaller learning rate to avoid catastrophic forgetting. The mid layers are updated with a rate of 10^{-6} and the bottom layers with a learning rate of 10^{-7}. An initial warm-up stage is added to the schedule that increases the learning rate gradually from 0 to 10^{-5} for 2 epochs. The experiment is re-run *twenty* times, each run uses a different random seed for parameter initialization and data shuffling. Fig. 5 shows the experimental results comparing F1 scores before and after adaptive training with varying training-set sizes. It can be seen from the results that risk-adaptive training achieves better scores than conventional training. This observation is consistent with the literature on domain-adaptation (Ganin & Lempitsky, 2015).

6.4. Out-of-distribution risk analysis

In the previous experiments we only dealt with the case when the test set comes from the same distribution as the training and validation sets. It is interesting to evaluate the proposed risk model's ability to detect mispredictions on out-of-distribution (OOD) data. In fact, the risk model itself is trained on the classifier's mistakes on the validation data, which reduces its performance in this scenario. However, making use of global metrics from external sources such as a knowledge base and incorporating a dynamic attention mechanism allow for more generalization ability. Table 2 shows the AUROC scores for the compared methods when using AIDA-CoNLL, CLUEWEB and WIKI datasets as source data sets and testing on MSNBC, ACE2004 and AQUAINT. In addition, for each of our training sets we use the

[4] https://dumps.wikimedia.org/enwiki/.

[5] https://github.com/attardi/wikiextractor.



(a) AIDA-CoNLL (b) WIKI (c) CLUEWEB

Fig. 5. Risk adaptive training comparative results. F1 scores on the test set are shown for different training set sizes. The shaded area around the curve represents the 95% confidence interval.

Table 2
Out-Of-Distribution risk analysis AUROC scores. The top 5 rows of each sub-table show the Area Under the ROC curve while the bottom 4 rows show the *p*-values between KBRisk and each alternative method for 20 runs across all OOD datasets.

AIDA-CoNLL	MSNBC	ACE2004	AQUAINT	WIKI	CLUEWEB
Baseline	0.8361	0.693	0.7499	0.7885	0.815
ConfidNet	0.5543	0.6479	0.5501	0.5532	0.5138
TrustScore	0.8462	0.7335	0.788	0.7316	0.7586
LearnRisk	0.7875	0.7543	0.8111	0.7622	0.8337
KBRisk	**0.8604**	**0.7866**	**0.8678**	**0.8136**	**0.842**
vs Baseline	0.001213*	1.51e−07*	1.519e−22*	9.203e−07*	1.26e−14*
vs ConfidNet	4.359e−09*	2.516e−07*	1.162e−12*	4.054e−10*	1.279e−09*
vs TrustScore	0.05301	5.214e−07*	1.514e−17*	5e−22*	7.285e−30*
vs LearnRisk	3.248e−12*	0.01529*	2.384e−13*	4.185e−12*	0.002003*
CLUEWEB	MSNBC	ACE2004	AQUAINT	AIDA-CoNLL	WIKI
Baseline	0.8336	0.6351	0.7177	**0.8142**	0.8292
ConfidNet	0.4023	0.611	0.4821	0.4683	0.3815
TrustScore	0.8351	0.652	0.7455	0.7051	0.7607
LearnRisk	0.8004	0.7581	0.8173	0.72	0.8115
KBRisk	**0.901**	**0.793**	**0.8351**	0.7181	**0.8349**
vs Baseline	1.109e−15*	2.005e−21*	5.174e−24*	1.78e−20*	0.02331*
vs ConfidNet	5.579e−19*	1.687e−16*	2.672e−22*	6.143e−12*	1.036e−18*
vs TrustScore	2.563e−17*	9.636e−20*	9.065e−17*	0.0001329*	2.541e−26*
vs LearnRisk	8.959e−19*	0.004097*	0.002609*	0.7472	1.223e−09*
WIKI	MSNBC	ACE2004	AQUAINT	AIDA-CoNLL	CLUEWEB
Baseline	0.803	0.6009	0.6843	0.7573	0.7116
ConfidNet	0.442	0.5766	0.515	0.5116	0.5072
TrustScore	0.8294	0.8345	0.7847	0.7413	0.7756
LearnRisk	0.8089	0.7219	0.7879	0.74	0.7284
KBRisk	**0.8608**	**0.8694**	**0.8919**	**0.7787**	**0.8357**
vs Baseline	0.001726*	1.407e−15*	1.263e−17*	0.3686	2.294e−12*
vs ConfidNet	1.219e−12*	1.107e−11*	3.717e−15*	3.658e−09*	9.318e−16*
vs TrustScore	0.1136	0.1114	9.528e−09*	0.1273	1.774e−05*
vs LearnRisk	0.0002292*	1.127e−09*	3.91e−12*	0.06792	1.439e−14*

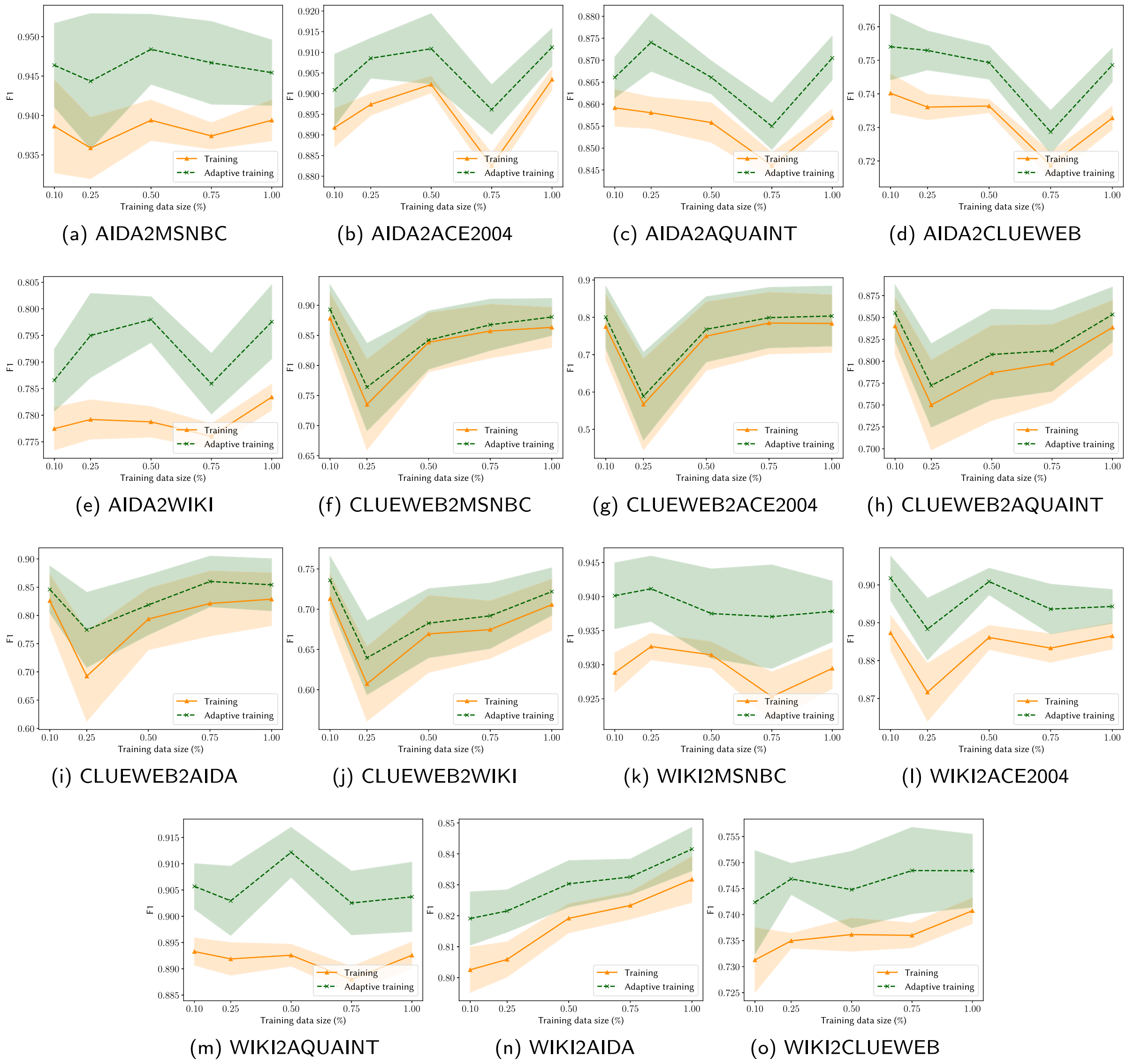
* *p*-values less than the confidence 0.05.

testsets of the remaining two sets as Out-Of-Distribution sets. We can see from the table that using AIDA-CoNLL or WIKI as source datasets, KBRisk achieves the best results on all OOD test sets. These results are mostly significant when using AIDA-CoNLL as source, as the t-test results between KBRisk and every other method show. In three out of five testsets, LearnRisk had the second best performance. When using WIKI as source, only for Trustscore, which ranks second best in multiple target sets, the statistical significance cannot be accepted for the confidence level of 0.05. As for other methods, KBRisk's performance is shown to be statistically significant. In the case of using CLUEWEB as source, KBRisk's AUROC score is only surpassed by the baseline on AIDA-CoNLL. Despite its simplicity, the baseline approach shows decent performance, especially when the target set was AIDA-CoNLL, which confirms the findings in Hendrycks and Gimpel (2017). LearnRisk had the second best performance in three testsets.

Given the results above, we also perform adaptive training on OOD testsets by using AIDA-CoNLL, WIKI and CLUEWEB as source datasets. The F1-scores comparing the conventional training with adaptive training are given in Fig. 6. We apply the same settings as in Section 6.4. It can be seen that despite the fluctuation in the results across training sizes, the adaptively-trained DNN can outperform the traditionally trained one on all OOD test sets.

6.5. Ablation study

In the conception of the risk model architecture, many design choices have been made. Some components had variants to choose from such as VaR versus CVaR or the choice of loss function. So, it is of great importance to validate these choices and quantify their influence on risk analysis results. We identify the following variations which we deem influential: the attention mechanism which makes use of entity embeddings to generate dynamic weight vectors per input instance. Modeling inter-dependence between risk features using a Multi-variate Normal random vector. Loss function definition using *pair loss* or *aggregate loss*. Table 3 shows the AUROC results on all test datasets



(a) AIDA2MSNBC (b) AIDA2ACE2004 (c) AIDA2AQUAINT (d) AIDA2CLUEWEB

(e) AIDA2WIKI (f) CLUEWEB2MSNBC (g) CLUEWEB2ACE2004 (h) CLUEWEB2AQUAINT

(i) CLUEWEB2AIDA (j) CLUEWEB2WIKI (k) WIKI2MSNBC (l) WIKI2ACE2004

(m) WIKI2AQUAINT (n) WIKI2AIDA (o) WIKI2CLUEWEB

Fig. 6. Risk adaptive training OOD comparative results. Figures (a)–(e), (f)–(j) and (k)–(o) show the F1 scores on MSNBC, ACE2004 and AQUAINT, AIDA and WIKI when AIDA, WIKI and CLUEWEB is used for training respectively. Scores on the test set are shown for different training set sizes. The shaded area around the curve represents the 95% confidence interval.

Table 3
Components ablation study. **LearnRisk** is the base model. **Att** represents the attention layer, **MN** represents the multi-variate normal layer, **CVaR** is the Conditional Value-at-Risk. **ALoss** is the aggregated loss while **PALoss** is the pairwise-aggregated loss combination. Values represent Area Under the ROC curve.

Dataset	AIDA-CoNLL	WIKI	CLUEWEB
LearnRisk	0.8230	0.8540	0.8057
+Att	0.8254	0.8602	0.8061
+MN	0.8375	0.8582	0.8139
+CVaR	0.8404	0.8558	0.8131
+ALoss	0.8533	0.8591	0.8149
+PALoss	**0.8553**	**0.8648**	**0.8161**

using different risk model component combinations. **LearnRisk** denotes the original raw risk model with a global weight vector and VaR risk measure but using the metrics proposed in Section 4.2, **Att** denotes the attention layer, **MN** denotes the multivariate normal layer, **CVaR** is the application of CVaR instead of VaR for risk estimation. **ALoss** and **PALoss** denote the aggregated loss (Ψ_ρ) and the sum of the pair loss and aggregated loss ($\Psi_s + \Psi_\rho$) respectively. The loss impact is specifically considered once all components are included, since in other variants, the used loss is the **PLoss** (Ψ_s).

It can be clearly observed on Table 3 that the components proposed in this work were beneficial in the overall performance across the tested datasets. The attention layer, being the component responsible for injecting KB knowledge via entity embeddings contributed in boosting the accuracy of the risk model on CLUEWEB. The multi-variate normal layer's contribution had a significant positive impact on AIDA-CoNLL and CLUEWEB datasets and the CVaR layer offered a slight improvement overall. **ALoss** had a big impact on performance compared with the default pair loss. Finally, it is obvious that **PALoss** converges to a better solution compared with **ALoss** on two out of three datasets, with CLUEWEB being the exception. This means that *pair loss* and *aggregate loss* are equally important to the final ranking. The final row in the table represents the full KBRisk model.

With a fixed risk model architecture, risk analysis effectiveness further relies on the quality of the rules extracted from the classifier's training data. The one-sided decision tree-based rule extraction algorithm and the training data available to it are predetermined. So, the quality of the rules is mainly dependent on the metrics that are used to generate them. Section 4.2 proposed three metric categories

Table 4
Metrics ablation study. **EMB** represents embedding-based metrics, **REP** represents DNN representation-based metrics and **KB** represents knowledge-base metrics. Values represent Area Under the ROC curve.

Source	#Rules	Dataset		
		AIDA-CoNLL	WIKI	CLUEWEB
EMB	130	0.7426	0.8052	0.7337
+REP	411	0.8268	0.8641	0.8043
+KB	2323	**0.8553**	**0.8648**	**0.8161**

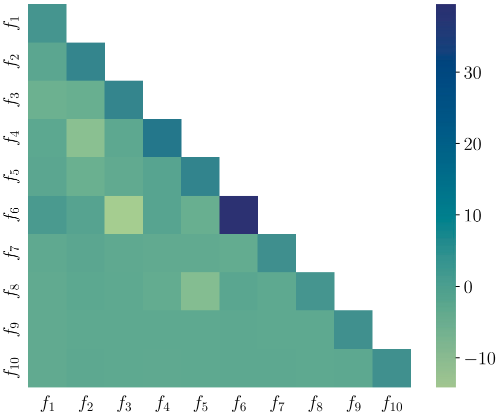


Fig. 7. Risk feature distribution's covariance matrix (Σ) example.

depending on the knowledge sources used to compute them. Therefore, we also evaluate their individual contributions to risk analysis. Table 4 shows AUROC results using various metric category combinations. The number of rules generated is also reported for each metric category combination. The *embedding*, *knowledge base* and *classifier representation* metric categories are denoted as **EMB**, **KB** and **REP** respectively. It is clear from the AUROC scores that each knowledge source adds to KBRisk's accuracy in detecting mispredictions. More specifically, the knowledge base extracted metrics lead to significant improvements.

6.6. Qualitative analysis

The solution depends on many qualitative characteristics that play a big role in ensuring its good performance. This section presents an analysis of these characteristics such as topic model coherence, entity coverage and entity evidence quality.

6.6.1. Risk estimation example

In this subsection, we illustrate through an example how the risk is measured in the case of KBRisk and how it compares to LearnRisk. We fitted an instance of KBRisk and an instance of LearnRisk on the DNN model mispredictions on AIDA-CoNLL dataset. We picked an example from AIDA-CoNLL test set and ran risk estimation from both risk model variants. Table 5 shows the inputs: the mention *World Series* within its context and the proposed candidates, the DNN model's prediction *Australian Tri-Series*, and the ground truth answer *World Series Cricket*. In addition, the table shows the intermediate scores for each variant consisting of: active risk features f_j and their corresponding static means μ_{f_j}, and learned weights w_j, the final mean μ_i and variance σ_i^2 for the distribution of π_i, and the calculated risk score ρ_i. The rank of the example according to ρ_i is also reported to be able to compare raw scores regardless of the score distribution. We limit the number of risk features to the top five features according to their weights w_j. In the case of KBRisk, we further plot in Fig. 7 a subset of the covariance matrix Σ containing only the active risk features for the example of Table 5. For more details on the metrics used in LearnRisk such as least_common_string and monge_elkan similarity metrics, refer to the original paper (Chen et al., 2020). For the metrics used by KBRisk, refer to Section 4.2. Note that KBRisk makes use of only one text similarity metric, which is edit_distance.

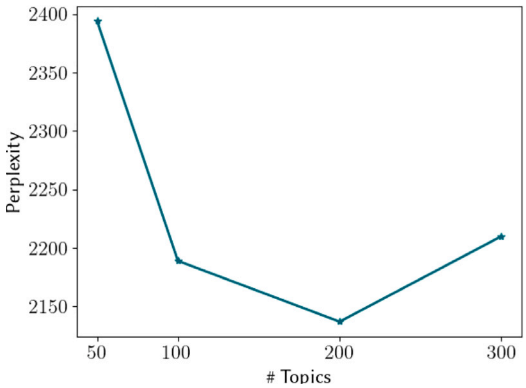


Fig. 8. Topic model perplexity for different number of topics.

The first observation is that LearnRisk, which uses string similarity metrics based only on the training data, has a smaller number of active features. Moreover, the DNN probability-based risk feature was among the top-5 active features. Meaning that for LearnRisk the output of the DNN contributed in the final estimation (with a weight of 0.05). On the other hand, KBRisk had more than five active risk features involving multiple metrics from various sources. This example represents a misprediction by the DNN model: the risk score should be relatively high in this case. Yet, looking at the rank of the example according to ρ_i for both LearnRisk and KBRisk, we see that LearnRisk estimated a lower risk than KBRisk. This is due to the high variance σ_i^2 causing the VaR score to be high even though both values for μ_i are low. KBRisk estimated lower variance by learning the covariance matrix between all feature distributions. Resulting in a better estimation for misprediction risk. We also notice that despite being ranked 1043rd by LearnRisk, the risk score ρ_i is still above 0.95. This is due to the VaR metric which tends to give values closer to the extremes. On the other hand, KBRisk, using CVaR, has better calibrated risk scores ρ_i.

6.6.2. Topic model coherence

The quality of the extracted topics depends on the number of documents used to fit the LDA model as well as the good choice of the number of topics n_T. Using wikipedia as a source, which contains millions of entities, ensures that the number of documents is not an issue. Fig. 8 shows the perplexity of the LDA model for different values of n_T. In our experiments, we set $n_T = 200$ as it represents a sweet spot.

6.6.3. Entity evidence quality

After the evidence is extracted, each entity will have one or more topics it belongs to as well as a set of keywords or tokens that are highly indicative of said entity. Table 6 shows some entities together with their corresponding evidence ranked by their scores: TF-IDF scores for keywords and LDA probabilities for topics. The topics shown in the table were labeled manually based on their top constituent words. They are shown in capital letters in the table. In all five examples, topics are among the top-5 features.

6.6.4. Dataset entity coverage

When using external knowledge, it is important to assess how much of the data at hand is covered by it. Table 7 shows the number of entities covered by the knowledge base for each dataset (validation and test sets). It clearly shows that more than 90% of the entities are covered on all datasets. Thanks to the incorporation of entity embeddings via the feature mapping matrix K, their risk feature weights are estimated from their position in the embedding space. I.e. novel entities are treated as their neighboring entities which were seen during training.

Table 5
Risk estimation comparative example between KBRisk and LearnRisk.

Mention		Cricket Australia Beat West Indies By Five Wickets Melbourne 1996 12 06 Australia beat West Indies by five wickets in a **World Series** limited overs match at the Melbourne Cricket Ground on Friday Scores West Indies 172 all out in 49 2 overs Shivnarine Chanderpaul 54 Australia 173 5 in 48 4 overs Greg Blewett 57 not out		
Candidates		World Series, 1994 World Series, 1944 World Series, World Series Most Valuable Player Award, Negro World Series, World Series Cricket, Australian Tri-Series, ATP International Series		
Model prediction		Australian Tri-Series		
Ground truth		World Series Cricket		
Model probability		0.1305		
LearnRisk	μ_i	0.1239		
	σ_i^2	0.081		
	ρ_i	0.9524		
	RANK(ρ_i)	1043/4486		
	Features	Expression	μ_{f_j}	w_j
		f_1 : least_common_string $> 0.3798 \wedge$ monge_elkan ≤ 0.6917	0.1208	0.66
		f_2 : $0.6913 \leq$ monge_elkan $\leq 0.7162 \wedge$ edit_distance > 0.4874	0.1388	0.18
		f_3 : monge_elkan ≤ 0.5249	0.1319	0.08
		f_4 : DNN	0.11	0.05
		f_5 : least_common_string > 0.4361	0.0686	0.03
KBRisk	μ_i	0.2930		
	σ_i^2	0.0082		
	ρ_i	0.8629		
	RANK(ρ_i)	183/4486		
	Features	Expression	μ_{f_j}	w_j
		f_1 : edit_distance ≤ 0.9128	0.1016	0.19
		f_2 : neg_class_dist $> 0.010 \wedge \hat{p}(e\|m) \leq 0.0714$ $\wedge$ topic_sim $> 0.4835 \wedge$ edit_distance > 0.4843	0.8631	0.16
		f_3 : pos_class_dist $\leq 0.1255 \wedge$ neg_class_dist > 0.0103 $\wedge \hat{p}(e\|m) \leq 0.2142 \wedge$ context_emb_sim_cos ≤ 0.1802	0.0906	0.09
		f_4 : pos_class_dist $\leq 0.1772 \wedge \hat{p}(e\|m) \leq 0.071$ $\wedge$ mention_emb_sim_euc $\leq 0.4899 \wedge$ edit_distance ≤ 0.1702 $\wedge$ context_text_sim ≤ 0.0233	0.8418	0.05
		f_5 : $0.2361 <$ context_text_sim $\leq 0.4642 \wedge$ edit_distance > 0.8248 $\wedge$ mention_emb_sim_euc ≤ 0.1798	0.1445	0.04

Table 6
Entity evidence examples. Top-5 tokens are shown per entity. Bold tokens represent topics and non-bold ones represent keywords.

Entity	Keyword/**Topic** (Probability)				
GPS signals	gps(0.62)	**software**(0.35)	satellites(0.29)	**space**(0.28)	**aerospace**(0.23)
Tony Mitchell (musician)	**music**(0.73)	porter(0.28)	band(0.26)	album(0.24)	singles(0.23)
Googleplex	google(0.84)	**software**(0.42)	california(0.16)	**structure**(0.12)	mountain(0.11)
Standard German	german(0.70)	**language**(0.42)	language(0.30)	dialects(0.29)	standard(0.23)
FDA (trade union)	fda(0.48)	**association**(0.35)	union(0.29)	secretary(0.19)	patel(0.18)

Table 7
Dataset entity coverage.

Dataset	Validation	Test
AIDA-CoNLL	90.28%	92.21%
WIKI	93.19%	95.61%
CLUEWEB	90.81%	93.66%

7. Conclusion

In this paper, we proposed a knowledge-based risk model for the task of ED. Leveraging external knowledge about real-world entities allowed for more accurate risk analysis. Moreover, we were able to perform adaptive deep learning by minimizing the misprediction risk on target data. Our extensive experiments on real benchmark data validated the efficacy of both the proposed risk analysis approach and the adaptive deep learning approach.

For future work, some limitations of the proposed risk analysis approach are worthy of further investigation. First, the proposed approach relies on task-specific metrics that are carefully designed. A future improvement may attempt to decouple the metrics-design phase from the specificities of the task or to devise an automatic way to generate the metrics. Second, for risk analysis, we assume the existence of a validation set with decent size, it is worth considering if risk model training can be reduced to a few-shot approach that can be fitted on only a handful of examples. Third, many newer models such as Large Language Models have already been pre-trained on large corpora including Wikipedia. It is interesting to explore how to leverage these large NLP models for risk analysis.

CRediT authorship contribution statement

Youcef Nafa: Conceptualization, Formal analysis, Writing – original draft. **Qun Chen:** Conceptualization, Supervision, Writing – review & editing. **Boyi Hou:** Methodology. **Zhanhuai Li:** Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data used in this work is openly accessible.

Acknowledgments

This work was supported by the Natural Science Foundation of China [grant numbers 62172335, 61732014, 61672432], and the Fundamental Research Funds for the Central Universities, China (3102019DX1004)

References

Al-Moslmi, T., Gallofré Ocaña, M., Opdahl, A. L., & Veres, C. (2020). Named entity extraction for knowledge graphs: A literature overview. *IEEE Access*, *8*, 32862–32881. http://dx.doi.org/10.1109/ACCESS.2020.2973928.

Alam, M., Buscaldi, D., Cochez, M., Osborne, F., Recupero, D. R., Sack, H., Sevgili, O., Shelmanov, A., Arkhipov, M., Panchenko, A., Biemann, C., Alam, M., Buscaldi, D., Cochez, M., Osborne, F., Refogiato Recupero, D., & Sack, H. (2022). Neural entity linking: A survey of models based on deep learning. *Semantic Web*, *13*, 527–570. http://dx.doi.org/10.3233/SW-222986.

Alhelbawy, A., & Gaizauskas, R. (2014). Graph ranking for collective named entity disambiguation. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (Volume 2: Short papers)* (pp. 75–80). Baltimore, Maryland: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P14-2013, URL: https://aclanthology.org/P14-2013.

Artzner, P., Delbaen, F., Eber, J. M., & Heath, D. (1997). *A characterization of measures of risk: Technical Report*, Cornell University Operations Research and Industrial Engineering.

Artzner, P., Delbaen, F., Eber, J. M., & Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, *9*, 203–228. http://dx.doi.org/10.1111/1467-9965.00068, URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9965.00068, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9965.00068.

Ayoola, T., Fisher, J., & Pierleoni, A. (2022). Improving entity disambiguation by reasoning over a knowledge base. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 2899–2912). Seattle, United States: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2022.naacl-main.210, URL: https://aclanthology.org/2022.naacl-main.210.

Bagga, A., & Baldwin, B. (1998). Entity-based cross-document coreferencing using the vector space model. In *COLING 1998 Volume 1: The 17th international conference on computational linguistics*. URL: https://aclanthology.org/C98-1012.

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Y. Bengio, & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference track proceedings*. URL: http://arxiv.org/abs/1409.0473.

Bates, S., Angelopoulos, A., Lei, L., Malik, J., & Jordan, M. I. (2021). Distribution-free, risk-controlling prediction sets. arXiv:2101.02703.

Bekkerman, R., & McCallum, A. (2005). Disambiguating web appearances of people in a social network. In *Proceedings of the 14th international conference on world wide web* (pp. 463–470). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/1060745.1060813.

Blei, D., Ng, A., & Jordan, M. (2001). Latent dirichlet allocation. In T. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems*. MIT Press, URL: https://proceedings.neurips.cc/paper/2001/file/296472c9542ad4d4788d543508116cbc-Paper.pdf.

Broscheit, S. (2019). Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)* (pp. 677–685). Hong Kong, China: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/K19-1063, URL: https://aclanthology.org/K19-1063.

Bunescu, R., & Paşca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In *11th conference of the european chapter of the association for computational linguistics* (pp. 9–16). Trento, Italy: Association for Computational Linguistics, URL: https://aclanthology.org/E06-1002.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on machine learning* (pp. 89–96). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/1102351.1102363.

Cao, Y., Hou, L., Li, J., & Liu, Z. (2018). Neural collective entity linking. In *Proceedings of the 27th international conference on computational linguistics* (pp. 675–686). Santa Fe, New Mexico, USA: Association for Computational Linguistics, URL: https://aclanthology.org/C18-1057.

Cao, N. D., Izacard, G., Riedel, S., & Petroni, F. (2021). Autoregressive entity retrieval. In *International conference on learning representations*. URL: https://openreview.net/forum?id=5k8F6UU39V.

Chen, Z., Chen, Q., Hou, B., Li, Z., & Li, G. (2020). Towards interpretable and learnable risk analysis for entity resolution. In *Proceedings of the 2020 ACM SIGMOD international conference on management of data* (pp. 1165–1180). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3318464.3380572.

Chen, Q., Chen, Z., Nafa, Y., Duan, T., Pan, W., Zhang, L., & Li, Z. (2022). Adaptive deep learning for entity resolution by risk analysis. *Knowledge-Based Systems*, Article 110118. http://dx.doi.org/10.1016/j.knosys.2022.110118, URL: https://www.sciencedirect.com/science/article/pii/S095070512201214X.

Corbière, C., Thome, N., Bar-Hen, A., Cord, M., & Pérez, P. (2019). *Addressing failure prediction by learning model confidence*. Red Hook, NY, USA: Curran Associates Inc..

Cucerzan, S. (2007). Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CONLL)* (pp. 708–716). Prague, Czech Republic: Association for Computational Linguistics, URL: https://aclanthology.org/D07-1074.

De Cao, N., Wu, L., Popat, K., Artetxe, M., Goyal, N., Plekhanov, M., Zettlemoyer, L., Cancedda, N., Riedel, S., & Petroni, F. (2022). Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, *10*, 274–290. http://dx.doi.org/10.1162/tacl_a_00460, URL: https://aclanthology.org/2022.tacl-1.16.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (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)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1423, URL: https://aclanthology.org/N19-1423.

Dredze, M., McNamee, P., Rao, D., Gerber, A., & Finin, T. (2010). Entity disambiguation for knowledge base population. In *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)* (pp. 277–285). Beijing, China: Coling 2010 Organizing Committee, URL: https://aclanthology.org/C10-1032.

Fang, W., Zhang, J., Wang, D., Chen, Z., & Li, M. (2016). Entity disambiguation by knowledge and text jointly embedding. In *Proceedings of the 20th SIGNLL conference on computational natural language learning* (pp. 260–269). Berlin, Germany: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/K16-1026, URL: https://aclanthology.org/K16-1026.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, *27*, 861–874. http://dx.doi.org/10.1016/j.patrec.2005.10.010, URL: https://www.sciencedirect.com/science/article/pii/S016786550500303X, rOC Analysis in Pattern Recognition.

Févry, T., FitzGerald, N., Soares, L. B., & Kwiatkowski, T. (2020). Empirical evaluation of pretraining strategies for supervised entity linking. CoRR abs/2005.14253. URL: https://arxiv.org/abs/2005.14253, arXiv:2005.14253.

Fleischman, M., & Hovy, E. (2004). Multi-document person name resolution. In *Proceedings of the conference on reference resolution and its applications* (pp. 1–8). Barcelona, Spain: Association for Computational Linguistics, URL: https://aclanthology.org/W04-0701.

Ganea, O. E., & Hofmann, T. (2017). Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2619–2629). Copenhagen, Denmark: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D17-1277, URL: https://aclanthology.org/D17-1277.

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning* (pp. 1180–1189). PMLR.

Globerson, A., Lazic, N., Chakrabarti, S., Subramanya, A., Ringgaard, M., & Pereira, F. (2016). Collective entity resolution with multi-focal attention. In *Proceedings of the 54th annual meeting of the association for computational linguistics (Volume 1: Long papers)* (pp. 621–631). Berlin, Germany: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P16-1059, URL: https://aclanthology.org/P16-1059.

Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 855–864).

Guo, Z., & Barbosa, D. (2018). Robust named entity disambiguation with random walks. *Semantic Web*, *9*, 459–479. http://dx.doi.org/10.3233/SW-170273.

Han, X., & Sun, L. (2011). A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 945–954). Portland, Oregon, USA: Association for Computational Linguistics, URL: https://aclanthology.org/P11-1095.

Han, X., & Sun, L. (2012). An entity-topic model for entity linking. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 105–115). Jeju Island, Korea: Association for Computational Linguistics, URL: https://aclanthology.org/D12-1010.

Han, X., & Zhao, J. (2009). Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on information and knowledge management* (pp. 215–224). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/1645953.1645983.

Hendrycks, D., & Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th international conference on learning representations*.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., & Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 782–792). Edinburgh, Scotland, UK: Association for Computational Linguistics, URL: https://aclanthology.org/D11-1072.

Hu, L., Ding, J., Shi, C., Shao, C., & Li, S. (2020). Graph neural entity disambiguation. *Knowledge-Based Systems*, *195*, Article 105620. http://dx.doi.org/10.1016/j.knosys.2020.105620, URL: https://www.sciencedirect.com/science/article/pii/S0950705120300861.

Huang, H., Heck, L. P., & Ji, H. (2015). Leveraging deep neural networks and knowledge graphs for entity disambiguation. CoRR abs/1504.07678. URL: http://arxiv.org/abs/1504.07678, arXiv:1504.07678.

Ji, H., & Grishman, R. (2011). Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 1148–1158). Portland, Oregon, USA: Association for Computational Linguistics, URL: https://aclanthology.org/P11-1115.

Jiang, H., Kim, B., Guan, M. Y., & Gupta, M. (2018). To trust or not to trust a classifier. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 5546–5557). Curran Associates Inc..

Kannan Ravi, M. P., Singh, K., Mulang', I. O., Shekarpour, S., Hoffart, J., & Lehmann, J. (2021). CHOLAN: A modular approach for neural entity linking on Wikipedia and Wikidata. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: Main volume* (pp. 504–514). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.eacl-main.40, URL: https://aclanthology.org/2021.eacl-main.40.

Kataria, S. S., Kumar, K. S., Rastogi, R. R., Sen, P., & Sengamedu, S. H. (2011). Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1037–1045). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2020408.2020574.

Kenter, T., Borisov, A., & de Rijke, M. (2016). Siamese CBOW: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th annual meeting of the association for computational linguistics (Volume 1: Long papers)* (pp. 941–951). Berlin, Germany: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P16-1089, URL: https://aclanthology.org/P16-1089.

Kolitsas, N., Ganea, O. E., & Hofmann, T. (2018). End-to-end neural entity linking. In *Proceedings of the 22nd conference on computational natural language learning* (pp. 519–529). Brussels, Belgium: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/K18-1050, URL: https://aclanthology.org/K18-1050.

Lazic, N., Subramanya, A., Ringgaard, M., & Pereira, F. (2015). Plato: A selective context model for entity resolution. *Transactions of the Association for Computational Linguistics*, *3*, 503–515. http://dx.doi.org/10.1162/tacl_a_00154, URL: https://aclanthology.org/Q15-1036.

Le, P., & Titov, I. (2018). Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long papers)* (pp. 1595–1604). Melbourne, Australia: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P18-1148, URL: https://aclanthology.org/P18-1148.

Le, P., & Titov, I. (2019). Boosting entity linking performance by leveraging unlabeled documents. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 1935–1945). Florence, Italy: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P19-1187, URL: https://aclanthology.org/P19-1187.

Ling, J., FitzGerald, N., Shan, Z., Soares, L. B., Févry, T., Weiss, D., & Kwiatkowski, T. (2020). Learning cross-context entity representations from text. URL: https://openreview.net/forum?id=HygwvC4tPH.

Malin, B., Airoldi, E., & Carley, K. M. (2005). A network analysis model for disambiguation of names in lists. *Computational & Mathematical Organization Theory*, *11*, 119–139. http://dx.doi.org/10.1007/s10588-005-3940-3.

Mann, G., & Yarowsky, D. (2003). Unsupervised personal name disambiguation. In *Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003* (pp. 33–40). URL: https://aclanthology.org/W03-0405.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In Y. Bengio, & Y. LeCun (Eds.), *1st international conference on learning representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop track proceedings*. URL: http://arxiv.org/abs/1301.3781.

Milne, D., & Witten, I. H. (2008). Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on information and knowledge management* (pp. 509–518). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/1458082.1458150.

Minkov, E., Cohen, W. W., & Ng, A. Y. (2006). Contextual search and name disambiguation in email using graphs. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 27–34). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/1148170.1148179.

Nie, F., Cao, Y., Wang, J., Lin, C. Y., & Pan, R. (2018). Mention and entity description co-attention for entity disambiguation. In *Proceedings of the thirty-second AAAI conference on artificial intelligence and thirtieth innovative applications of artificial intelligence conference and eighth AAAI symposium on educational advances in artificial intelligence*. AAAI Press.

Pedersen, T., Purandare, A., & Kulkarni, A. (2005). Name discrimination by clustering similar contexts. In A. Gelbukh (Ed.), *Computational linguistics and intelligent text processing* (pp. 226–237). Berlin, Heidelberg: Springer Berlin Heidelberg.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (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)* (pp. 2227–2237). New Orleans, Louisiana: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N18-1202, URL: https://aclanthology.org/N18-1202.

Ratinov, L., Roth, D., Downey, D., & Anderson, M. (2011). Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 1375–1384). Portland, Oregon, USA: Association for Computational Linguistics, URL: https://aclanthology.org/P11-1138.

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *The Journal of Risk*, 21–41.

Sen, P. (2012). Collective context-aware topic models for entity disambiguation. In *Proceedings of the 21st international conference on world wide web* (pp. 729–738). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2187836.2187935.

Sevgili, Ö., Panchenko, A., & Biemann, C. (2019). Improving neural entity disambiguation with graph embeddings. In *Proceedings of the 57th annual meeting of the association for computational linguistics: Student research workshop* (pp. 315–322). Florence, Italy: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P19-2044, URL: https://aclanthology.org/P19-2044.

Sevgili, Ö., Shelmanov, A., Arkhipov, M. Y., Panchenko, A., & Biemann, C. (2020). Neural entity linking: A survey of models based on deep learning. CoRR abs/2006.00575. URL: https://arxiv.org/abs/2006.00575, arXiv:2006.00575.

Shahbazi, H., Fern, X. Z., Ghaeini, R., Obeidat, R., & Tadepalli, P. (2019). Entity-aware elmo: Learning contextual entity representation for entity disambiguation. CoRR abs/1908.05762. URL: http://arxiv.org/abs/1908.05762, arXiv:1908.05762.

Smith, S. L., Kindermans, P. J., & Le, Q. V. (2018). Don't decay the learning rate, increase the batch size. In *International conference on learning representations*. URL: https://openreview.net/forum?id=B1Yy1BxCZ.

Tardivo, G. (2002). Value at risk (var): The new benchmark for managing market risk. *Journal of Financial Management & Analysis*, *15*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Williamson, R., & Menon, A. (2019). Fairness risk measures. In K. Chaudhuri, & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 6786–6797). PMLR, URL: https://proceedings.mlr.press/v97/williamson19a.html.

Xiao, Y., Pei, Q., Yao, L., & Wang, X. (2020). Recrisk: An enhanced recommendation model with multi-facet risk control. *Expert Systems with Applications*, *158*, Article 113561. http://dx.doi.org/10.1016/j.eswa.2020.113561.

Yamada, I., Shindo, H., Takeda, H., & Takefuji, Y. (2017). Learning distributed representations of texts and entities from knowledge base. *Transactions of the Association for Computational Linguistics*, *5*, 397–411. http://dx.doi.org/10.1162/tacl_a_00069, URL: https://aclanthology.org/Q17-1028.

Yamada, I., Washio, K., Shindo, H., & Matsumoto, Y. (2022). Global entity disambiguation with BERT. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 3264–3271). Seattle, United States: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2022.naacl-main.238, URL: https://aclanthology.org/2022.naacl-main.238.

Yang, X., Gu, X., Lin, S., Tang, S., Zhuang, Y., Wu, F., Chen, Z., Hu, G., & Ren, X. (2019). Learning dynamic context augmentation for global entity linking. 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)* (pp. 271–281). Hong Kong, China: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D19-1026, URL: https://aclanthology.org/D19-1026.

Zhang, P., Wang, J., Farhadi, A., Hebert, M., & Parikh, D. (2014). Predicting failures of vision systems. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Zheng, Z., Li, F., Huang, M., & Zhu, X. (2010). Learning to link entities with knowledge base. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics* (pp. 483–491). Los Angeles, California: Association for Computational Linguistics, URL: https://aclanthology.org/N10-1072.

Youcef Nafa is a Ph.D student in the School of Computer Science in Northwestern Polytechnical University. His research interests include deep learning and artificial intelligence.

Qun Chen is a professor in the School of Computer Science in Northwestern Polytechnical University. His current research interests include gradual machine learning and risk analysis for AI.

Boyi Hou received Ph.D in School of Computer Science in Northwestern Polytechnical University. His research interests include data quality and artificial intelligence.

Zhanhuai Li is a professor in the School of Computer Science in Northwestern Polytechnical University. His research interests include data storage and management. He has served as Program Committee Chair or Member in various conferences and committees.