Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 1201–1208
Marseille, 20-25 June 2022

Development of a Benchmark Corpus to Support Entity Recognition in Job Descriptions

Thomas AF Green, Diana Maynard, Chenghua Lin
University of Sheffield / Regent Court
211 Portobello
Sheffield S1 4DP, UK
{tafgreen1, d.maynard, c.lin}@sheffield.ac.uk

Abstract

We present the development of a benchmark suite consisting of an annotation schema, training corpus and baseline model for Entity Recognition (ER) in job descriptions, published under a Creative Commons license. This was created to address the distinct lack of resources available to the community for the extraction of salient entities, such as skills, from job descriptions. The dataset contains 18.6k entities comprising five types (Skill, Qualification, Experience, Occupation, and Domain). We include a benchmark CRF-based ER model which achieves an F_1 score of 0.59. Through the establishment of a standard definition of entities and training/testing corpus, the suite is designed as a foundation for future work on tasks such as the development of job recommender systems.

Keywords: entity recognition, corpus development, job descriptions, natural language processing

1. Introduction

The identification and extraction of salient entities is an important task in many real-world information extraction applications such as text classification, efficient search algorithms, and content recommendations (Li et al., 2020). In recruitment, job-seekers and recruiting companies alike benefit from systems that automatically and continuously acquire up-to-date information about listed job roles and applicant profiles in terms of skills, qualifications, and experience.

In addition, these communities would benefit from investigation into the gap between candidate skills and open positions. This requires tools that can automatically identify and extract skills and related entities from unstructured text data.

However, the development of Entity Recognition (ER) models to perform these tasks is severely hindered by the lack of publicly available training data. Many available ER corpora consist of general news articles (Lawson and Eustice, 2010), while information about job descriptions is typically only available on online job portals. Skills, which have no agreed definition in the literature (see Section 2), are often general noun phrases rather than the proper names typically associated with Named Entities, making them harder to detect using gazetteer-based approaches. We define skills explicitly in Section 3.1.

To develop better job matching tools, we need to address these problems. We first establish a definition of the relevant entities in order to guide the collection of human-labelled data to be used for training and evaluation of automatic ER tools. Building on existing frameworks (Khobreh et al., 2016), we developed our schema, over several iterations, to include five distinct entities: Skills, Qualifications, Experiences, Domains, and Occupations (see Section 2 for more detail), and use this to build a corpus of annotated UK job descriptions. We then present a benchmark ER model using CRF architecture, which is also freely available and can be used as a baseline. Source code can be found in the associated repository.

An ER system trained on this data could then be used to compile the input to a job recommendation system, which, given suitable training data of matched candidate profiles (e.g. CVs, LinkedIn profiles) and job descriptions, is able to recommend jobs to candidates and vice versa based on the set of skills the candidates have and that jobs require. In addition, the work in this paper could be of use when investigating the current job climate in terms of the skills that jobs require and candidates possess, or investigating how skills (or demand for skills) change over time.

The core contributions are thus as follows:

- A list of entity classifications and their definitions in the form of an annotation schema for salient entities within job descriptions, made publicly available
- A public, labelled dataset for the development and evaluation of ER systems
- A benchmark ER system trained on this data

2. Related Work

In traditional machine learning approaches, 'feature extraction' refers to the process of building derived values ('features') from initial data to facilitate the subsequent learning and model establishment steps. In the context of job recommendation, this involves parsing unstructured text and extracting salient details from applicant profiles or job descriptions that are used as input to a recommendation model.

Solutions for matching between applicant profiles and job descriptions tend to use the 'skills' contained in the input data as the features to be extracted and used as input to a recommendation model (Almalis et al. (2014); Choudhary et al. (2016); Hoang et al. (2018); Gugnani and Misra (2020)). The underlying assumption is that a high similarity between the set of skills of an applicant and the set of skills required for a job is a strong indicator of a good fit. However, there are two main issues in existing literature regarding skill extraction for feature selection. Firstly, there is no academic consensus on the definition of a skill, which makes the comparison of different extraction methods a difficult task. Secondly, it is unclear (or relatively unexplored) which types of methods would give strong performance for skill extraction from text, so there is a need for research into the evaluation of different methods.

Regarding the lack of consensus on skill definition, related work into skill extraction falls mainly into one of two groups; the first omits a formal definition of a skill and leverages some other feature of the available data for identification, and the second refers to a public database of skills which are used as the terms for extraction.

An example of the first group is the work by Bastian et al. (2014), which allowed the users of a service (LinkedIn) to define skills themselves without explicit guidance from the researchers nor a formal definition. Other work assumes that anything contained in a user-defined 'Skills' section of an applicant profile qualifies as a skill (Maheshwari et al. (2010); Kivimäki et al. (2020); Karakatsanis et al. (2017)), which tends to introduce noise in the extraction. In some cases, 'field experts' are employed to annotate terms such as skills within job descriptions, and terms with high inter-annotator agreement are classified as skills (Gugnani and Misra, 2020). The limitation of these methods is that, without a formal definition, they are not reproducible outside of their specific contexts; by restricting the environment for the detection of skills to an explicitly defined Skills section in an applicant profile, for example, skills referred to in other sections will be missed, and in cases where the format of the profile omits a Skills section entirely, these methods will perform poorly.

The second group refers to a public database of skills such as O*NET[1] or those defined in the official frameworks such as the European Qualifications Framework (EQF), part of the European Skills/Competences, Qualifications and Occupations commission (ESCO; Khobreh et al. (2016)). The main limitation of using public databases of skills is that they are not effective for detecting new skills or detecting known skills expressed in new ways, and require constant updating in order to retain their usefulness. In areas of industry that feature constant development of new techniques, such as machine learning or computer programming, new methods and techniques will elude skill databases until they have been identified by the database maintenance teams and added. For example, an applicant profile may state proficiency in '*onboarding new hires*', referring to their skill in mentoring new employees. Although sections include teaching and training, coaching and mentoring, the term '*onboarding*' does not appear in O*NET nor ESCO skill databases, and skill extraction methods using these databases would be unable to detect this skill. Also, while ESCO is updated with new terms annually[2], skill extraction methods using this database could still be up to a year out of date. Our proposed method addresses these gaps by providing a schema for defining skills and related entities. Additionally, this schema is used to collect a human-labelled dataset of skills and related entities in job descriptions, which can be used to train an ER system for automatic detection.

Although skill classifiers have been developed, they are not suitable for comparison due to the differing definitions of skills. For example, Hoang et al. (2018) present the SKILL system which includes parts of job titles in their detection (e.g. '*financial*' in '*financial accountant*') and excludes other terms that our schema defines as skills (e.g. in the phrase '*monitoring budgets, developing forecasts, and investigating variances*', only the terms '*budgets*' and '*forecasts*' are classified as skills, whereas our schema would identify '*monitoring budgets*', '*developing forecasts*', and '*investigating variances*' as skills - see Section 3.1). Our hypothesis is that starting with a wider variation of terms will result in better matching when skills are extracted as features for job recommendation.

Moreover, related work focuses on 'skills' for extraction, and tends not to extend the scope to the extraction of related entities. We theorise that related entities may be useful in a job recommendation system, such as 'domains' (as exposure to a particular domain may be beneficial for roles in the same domain), 'occupations' (since acting in a particular job role may be suitable for certain types of jobs), and 'experience' (which may be used to quantify the proficiency of a candidate regarding a particular skill or occupation). Our schema includes definitions for these entities and they are included in our benchmark ER system.

3. Annotation Task Description

3.1. Schema of Entity Types

The annotation schema was defined through an iterative process of performing the annotation task in conjunction with reusing and adapting definitions from previous work (Gugnani and Misra, 2020; Shi et al., 2020; Hoang et al., 2018), the European Qualifications Framework (Khobreh et al., 2016), as well as advice from an HR Generalist working with Recruitment Software company TribePad[3] who volunteered to take

[1] https://www.onetonline.org/

[2] https://tinyurl.com/ESCO-v1

[3] https://tribepad.com/

part in a pilot annotation task. For example, while the EQF marks the distinction between 'Knowledge', 'Skills', and 'Attitudes', annotators in early task iterations were largely unable to differentiate between even 'Hard Skills' and 'Soft Skills' in practice, for example, the Hard Skill '*familiarity with European Standards*' was misclassified as a Soft Skill by 40% of annotators, and the Soft Skill '*maintaining high levels of accuracy*' was misclassified as a Hard Skill by 40% of annotators. Consequently, the 'Hard Skill' and 'Soft Skill' classes were collapsed into one all-encompassing 'Skill' classification.

The descriptions of entities in our annotation schema are summarised in Table 1, and the full version of the schema and annotation guidelines presented to Amazon Mechanical Turk (AMT) workers during data collection is contained in the repository along with the labelled corpus[4].

During initial development, the pool of annotators was restricted to 20 individuals with no prior experience of entity annotation tasks, and rounds of testing consisted of a random sample of annotators completing an annotation task with incremental changes in order to optimise inter-annotator agreement. Changes included the design and functionality of the annotation platform, the class distinctions themselves (including the combination of initially defined 'Hard Skill' and 'Soft Skill' classifications), and the structure of the annotation guidelines, which initially included only the list of entity classifications and their definitions, but was expanded to include a series of user-friendly 'clarification questions' in FAQ format as well as worked examples of annotated job descriptions.

3.2. Corpus

Our corpus of job descriptions came from the publicly available Kaggle dataset[5]. No original source for these is listed, but they appear to have been scraped from online job portals such as TotalJobs[6] and are limited to positions within the UK. A wide variety of industries are represented, such as IT, Finance, Healthcare, and Sales. After removing all html formatting and invalid UTF-8 code units, and splitting job descriptions into sentences[7] , the data consisted of 4,917,794 items (sentences). We randomly sampled 10,000 items for annotation, and a further 20 items to form the qualification set. We manually annotated a further 586 items to form the gold standard for both manual annotation and model evaluation.

4. HIT Design

To make the annotation task more convenient for AMT Workers, a customized user interface was used and detailed annotation guidelines were provided. Both the qualification task and the live annotation task were compensated, at $0.04/HIT and $0.08/HIT respectively, the latter equating to the standard minimum wage in the country in which the task was deployed.

Annotation guidelines presented to Workers included a full description of the annotation schema including examples of each class, as well as an FAQ section which clarified all Worker questions that arose during task development. In addition, a set of 8 worked examples was included, showing a fully annotated work item with explanations detailing which entities had been labelled, and the reasoning behind each classification.

5. Criteria for Worker Qualification

Workers were required to pass a 'qualification task' before they were assigned a bespoke qualification allowing them to contribute to the live task. To be eligible to work on the qualification task, they were required to be demonstrably competent at completing tasks on the platform; specifically, to have completed and had approved more than 5,000 HITs on the AMT platform and have achieved a lifetime approval rate of greater than 95%.

The qualification task featured 20 HITs for which the gold standard was available. Workers were encouraged to read the instructions carefully and complete as many HITs as possible. However, there were a variety of reasons why some Workers did not complete all HITs, such as the disinclination to commit too much time to a task for which, from their perspective, there is no guarantee of compensation (McInnis and Leshed, 2016). This was taken into account when calculating the threshold for Worker qualification.

178 individual Workers contributed to the qualification task with varying levels of accuracy and completeness. There is no established threshold of accuracy for Worker acceptance in related literature. Although greater accuracy of Worker annotations versus gold standard will lead to greater resultant model performance, a high threshold will result in fewer Workers eligible for contribution, leading to slower data collection rates.

To investigate the ideal threshold of accuracy to require of Workers, experimentation was performed using a standard ER dataset; the CoNLL-2003 Shared Task: Language Independent Named Entity Recognition (F. Tjong Kim Sang and De Meulder, 2003). The premise of this investigation was that recently developed ER models are able to learn directly from noisy human annotations, eliminating the need for label aggregation (Rodrigues and Pereira, 2017), and that examining the relationship between Worker performance (varied by artificially inducing noise) and resultant model performance may yield an appropriate threshold to

[4] https://tinyurl.com/skill-extraction-dataset

[5] https://tinyurl.com/trainrev1

[6] https://www.totaljobs.com

[7] Early testing suggested annotators performed poorly when items were too long. Splitting job descriptions into sentences improved accuracy on the annotation task and little was lost in terms of context when doing so.

Entity Name	Brief Description	Examples
Skill	Tasks that can be performed, or attributes and abilities (including soft skills) that enable people to perform tasks	*computer programming*, *French*, *honesty*
Qualification	Official certifications obtained through taking a course or passing an exam or appraisal	*Bachelor's Degree*, *chartership*, *three A-levels*
Experience	Lengths of time relating to a position or skill	*2 years experience*, *minimum of 5 years experience*
Occupation	Job titles, including abbreviations and acronyms	*Teaching Assistant*, *CEO*, *Chief Executive Officer*
Domain	Areas of industry in which someone might have knowledge or experience	*aerospace*, *oil industry*, *education*

Table 1: A brief description of entities for annotation. Full details can be found in the repository.

require of Workers for admitting them to contribute to our corpus.

Two distinct types of noise were investigated based on the cause of annotator misclassification: 'random' noise, where annotators make random errors (i.e. where any (incorrect) classification is equally likely to be selected); and 'systematic' noise, where annotators make consistent errors (by consistently misclassifying class A as class B). We also investigated the effect of reducing noise by artificially correcting annotated labels to simulate higher performance. This method could only simulate 'random' de-noise, but gives us an idea of the effect of annotators performing better than their current rate, i.e. if we were to increase the minimum accuracy level required.

We induced both forms of noise and random de-noise from proportions of 0 to 1 in increments of 0.02 and observe linear relationships between noise proportion and average Worker performance (% accuracy). A separate model was trained on each noised set of CoNLL training data using a Convolutional Neural Network with CrowdLayer proposed by Rodrigues and Pereira (2017). Code for reproducing this is publicly available[8].

Model F_1 is shown at varying levels of Worker accuracy in Figure 1. We observe a lower threshold of Worker performance at around 40% Worker accuracy, below which resultant model performance is poor ($<$ 50 model F_1). This is especially prevalent when inducing systematic noise (see Figure 1b).

Additionally, there seems to be a slight increase of model performance at around 70% Worker accuracy, which guided our decision to use this as our threshold. We also required Workers to have annotated at least 100 tokens in order to reasonably evaluate their performance.

39 Workers achieved an accuracy of greater than 70% on the qualification task and had annotated more than 100 tokens, and consequently only these Workers were invited to contribute to the live task.

6. Data Analysis

Sentences	10,000
Tokens	245,606
Avg. tokens per sentence	24.6
Annotation spans (post aggregation)	18,617
Annotated tokens (post aggregation)	79,826
Avg. tokens per annotation	4.3
Number of independent Annotators	25

Table 2: Annotated corpus statistics.

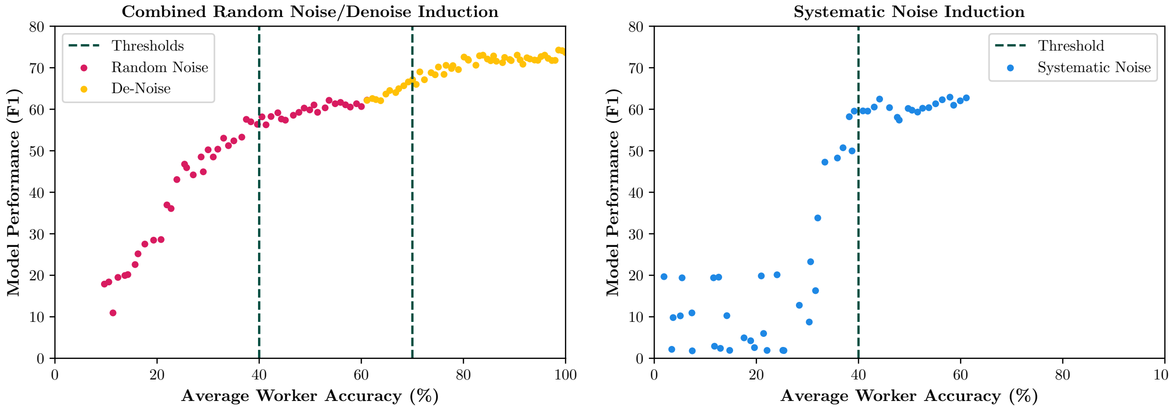
Label	Frequency	Proportion
Skill	66,732	28.56%
Occupation	6,117	2.62%
Domain	3,705	1.59%
Experience	1,328	0.57%
Qualification	1,944	0.83%
None	153,802	65.83%
Total	**233,628**	

Table 3: Class distribution for the live, aggregated corpus (one label per token).

Label	Frequency	Proportion
Skill	2,136	25.19%
Occupation	306	3.61%
Domain	100	1.18%
Experience	29	0.34%
Qualification	68	0.80%
None	5,839	68.87%
Total	**8,478**	

Table 4: Class distribution for the test set.

[8] https://tinyurl.com/noise-induction-experiments



(a) Combined Random Noise/De-noise Induction

(b) Systematic Noise Induction

Figure 1: Graph to show the relationship between average Worker accuracy (%) and resultant trained model F_1 after artificially inducing and removing noise in Worker annotations.

6.1. Size and Distribution

Table 2 lists general statistics of the annotated corpus, and Table 3 shows the distribution of class labels in the annotated corpus after aggregation to yield one label per token (see Section 7.1 for details regarding label aggregation). Similarly, Table 4 shows the distribution of class labels for the test set generated by the author of the annotation schema. We observe a similar distribution in both corpora.

6.2. Inter-Annotator Agreement (IAA)

Although Cohen's κ (Equation 1) is the standard measure of IAA, there have been several issues raised regarding its application in entity annotation tasks (Hripcsak and Rothschild, 2005) especially in cases where class distribution is unbalanced and where un-annotated tokens are much more common than annotated tokens. In these cases, Cohen's κ is calculated twice under two separate conditions: evaluating all tokens in the data, and evaluating only the annotated tokens in the data.

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{1}$$

Typically, including 'None' labels from calculation would show an inflated value of κ since the 'None' label is by far the most prevalent, and the high frequency of cases in which neither annotator has labelled a token tends to raise the observed agreement level. However, this is not the case in our data. Distribution of Worker contribution is non-uniform and non-normal, and the intersection of work between the majority of worker pairs is small ($<$ 10 sentences or $<$ 250 tokens). Since κ is calculated between each pair of annotators that contributed to at least one shared item and averaged across all pairs, there are several pairs of annotators that show an indeterminate κ agreement; if both annotators in a given pair have identified no entities across all reviewed tokens, the expected agreement p_e will be equal to 1, and κ will be indeterminate with a denominator of 0. For the κ statistics shown here, a case of indeterminate kappa between annotator pair $\{i, j\}$ is interpreted as perfect agreement ($\kappa_{ij} = 1$).

Pairwise F_1 on annotated tokens only has been suggested as a better measure for agreement in ER tasks (Deleger et al., 2012). We thus compute the micro F_1 on annotated tokens as the focal method of IAA, but Cohen's κ and Krippendorff's α statistics are provided to give additional insight (see Table 5).

Cohen's κ on all tokens	0.49
Cohen's κ on annotated tokens only	0.73
Krippendorff's α	0.55
F_1 on annotated tokens only	0.90

Table 5: IAA on the live corpus, calculated by averaging pairwise comparisons between all combinations of annotators where both annotators labelled a shared item.

7. Data Preprocessing

All entities were labelled using the BIO scheme. Although error is inevitable in human labelling tasks, it is feasible to mitigate some aspects. Preliminary analysis suggested that there were three sources of noise that could be mitigated prior to model training (referred to here as 'preprocessing'): label aggregation; reclassification of 'Experience' spans; and splitting multi-term spans.

Postprocessed data is included alongside raw data in the public repository associated with this research paper.

7.1. Label Aggregation

There are several established methods of label aggregation, such as majority agreement, simply removing items containing disagreements, or probabilistic aggregation methods in which annotators are identified as ‘trustworthy’ or otherwise on gold-standard tasks and weighting their annotations accordingly (Hovy et al., 2013). Alternatively, rather than extracting the single objective classification for each entity through agreement resolution methods, it is possible to learn a classifier directly from the annotations by assigning a distribution score to each label (Rodrigues and Pereira, 2017).

Since each token is annotated by two independent Workers, a simplification of the method of Hovy et al. (2013) was used for disagreement, where labels were assigned preferentially from higher-performing Workers inferred from qualification task results.

7.2. Reclassification of ‘Experience’ Spans

Preliminary analysis yielded a number of insights. According to the schema, ‘Experience’ spans must be quantified by length of time (e.g. ‘*2 years experience*’. A number of spans classified as Experience did not meet this criteria (e.g. ‘*experience managing clients*’), but did meet the criteria for the ‘Skill’ classification.

A ‘re-classification’ step was therefore added to the preprocessing pipeline in order to identify and correct these errors. Regular expression and inflect[9] Python packages were utilised to identify all spans that did not contain an expression of time (in word or number form) and reclassify the entire span from ‘Experience’ to ‘Skill’. This reduced the number of Experience spans from 239 to 144 (40% reduction), which were manually checked. No other classes were affected.

7.3. Splitting Multi-term Spans

A second finding from preliminary analysis was that annotators tended not to split lists of entities into separate spans, choosing instead to identify everything included in the list as one single span of the relevant entity type. For example, the sequence ‘*Asbestos Surveyors, Lead Asbestos Surveyors, Asbestos Analysts*’ was annotated as one single entity, whereas this should be three distinct entities with commas denoting the boundaries.

The correct splitting of entities is important for our task for two reasons. Firstly, it represents an issue for model training, in that if the training data does not reflect the correct distinction between multiple consecutive entities of the same type, it is unlikely that the resultant model will be able to, and will achieve poor performance when evaluated on the test set which features accurate entity separation.

Secondly, the intended use of a system trained to identify and extract entities from job descriptions is for feature extraction in a larger system developed to match applicant profiles and job descriptions. For this purpose, it is important that entities are discrete to ensure that each are evaluated independently to more accurately represent the requirements of a job from its description or an applicant from their profile. All instances of punctuation were re-classified with the ‘None’ label, and in cases where this split an annotated span, the following tokens became the start of a new span. Affected items were then manually checked to ensure legibility. The class distribution for the data after the preprocessing steps is shown in Table 6.

Label	**Frequency**	**Proportion**
Skill	65,632	28.09%
Occupation	5,964	2.55%
Domain	3,628	1.55%
Experience	800	0.34%
Qualification	1,716	0.73%
None	155,888	66.72%
Total	**233,628**	

Table 6: Class distribution for the preprocessed data (one label per token).

8. Baseline CRF Model

8.1. Settings

Conditional Random Fields (Lafferty et al., 1999) are commonly applied to structured prediction tasks such as ER to model structural dependencies, and present an appropriate benchmark setting for entity extraction. The output sequence is modelled as the normalised product of the feature function. Its formula is shown in Eq. 2, where X is the set of input vectors, y_i is the label at data point i, $Z(X)$ is the normalisation, and λ is the learned feature function weights.

$$p(y|X,\lambda) = \frac{1}{Z(X)} \exp \sum_{i=1}^{n} \sum_{j} \lambda_j f_i(X, i, y_{i-1}, y_i) \tag{2}$$

The NLTK[10] method of feature preparation was used, and the CRF model was trained over 100 epochs using L1 and L2 regularization coefficients found during parameter optimisation through Randomized Search. Results for the CRF are shown in Table 7.

8.2. Error Analysis

We observed instances of errors in classification from the baseline CRF model and have diagnosed likely sources.

8.2.1. Specific vs. General Applications of Skills

Our annotation schema states that, when a Skill is applied to a particular task, the details of the task should only be contained in the skill-term if it is a specific application (e.g. ‘*creative technical documentation*’) and not a general application (e.g. ‘*cleaning kitchens*’, where only ‘*cleaning*’ should be classified as a Skill). The CRF model is largely unable to distinguish between specific and general applications, and tends to include the application in either case. Examples of this are

[9] https://github.com/jaraco/inflect

[10] https://www.nltk.org/

Label	P	R	F_1	Support
B-Skill	0.69	0.37	0.48	676
I-Skill	0.53	0.71	0.61	1429
B-Qualification	0.72	0.50	0.59	26
I-Qualification	0.39	0.23	0.29	40
B-Occupation	0.90	0.65	0.75	137
I-Occupation	0.93	0.71	0.81	164
B-Experience	0.86	0.67	0.75	9
I-Experience	0.42	0.76	0.54	17
B-Domain	0.53	0.40	0.46	60
I-Domain	0.34	0.28	0.31	39
micro avg	0.58	0.60	0.59	2597
macro avg	0.63	0.53	0.56	2597
weighted avg	0.61	0.60	0.58	2597

Table 7: Results for CRF model (trained on preprocessed data). Precision, Recall, and F_1-Score are presented.

shown below, with the general application of the skill in parentheses, where the model incorrectly treats all tokens in each example as part of a classified span:

- training and developing new members (of the brigade)
- leading continuous improvement in business operations (with attention to our warehouse team and suppliers)

8.2.2. Multi-entity Span Classifications

As part of data preprocessing, large annotated spans that contain multiple discrete entities were split by punctuation (see Section 7.3). However, the CRF model often fails to split entities appropriately, and includes multiple entities of the same entity type within one span. This is true in particular of the Skills class, and contributes to the poor recall of the 'B-Skill' label (see Table 7). Examples of this are shown below, where the CRF model has identified the entirety of each example as one span, but the correct divisions are notated by parentheses:

- (communication) and (influencing) skills, ability to (embrace and apply leading practice tools and techniques), proven (customer service) orientation and (collaborative) approach
- (respond to internal and external stakeholder queries) in a timely manner and (proactively seek to resolve stakeholder issues)

8.2.3. Implications and Solutions

These two sources of error appear to be failures of the CRF model caused by an inability to correctly *terminate* an identified span. If the entities were used as features for a job recommendation system, these limitations would have the effect of reducing the number of features, which might present an issue for some recommendation algorithms (e.g. a bipartite graph matching approach).

A potential solution to these issues would be to use contextualised word embeddings (Turney and Pantel, 2010), which assign each token a single vector based on its context and, to some extent, capture the semantics of the word. An ER model that takes the semantics of words into account may be better able to distinguish between specific and general applications of skills, and may be better suited to identifying sensible termination points for spans to prevent multi-entity span classifications.

9. Ethical Considerations

The main ethical consideration for this research is the use of crowdsourcing data. Sabou et al. (2014) raise three issues regarding the use of crowdsourcing: how to acknowledge contributions; how to ensure contributor privacy and well-being; and how to deal with consent and licensing issues.

Since data was crowdsourced through the AMT platform, Workers were anonymised through the use of a unique Worker ID, and their details were restricted (with the exception of general statistics regarding their past performance on the platform, and the general location e.g. 'EU West').

To ensure Worker well-being, all contributions were compensated at a rate equivalent to UK minimum wage (at time of data collection).

Finally, the dataset is published under a 'no rights reserved' Creative Commons BY license, allowing for commercial and academic use of the data with attribution.

10. Conclusion and Future Work

In this paper, we have presented a new corpus for ER in the recruitment domain, annotated with five entity types. These types are not available in standard Named Entity Recognition corpora, but are the most relevant to this domain for tasks such as job recommendation. The data presented in this paper provides an ideal training set for this task, and is a suitable size for fine-tuning a pretrained model.

Additionally, we have presented an annotation schema to facilitate the collection of additional data, and a baseline CRF model for entity extraction, and have suggested methods for schema development, task construction, and corpus creation. All resources associated with this paper are made publicly available[11] under a Creative Commons BY license. Included in these resources is a Datasheet (Gebru et al., 2018) that describes the data and its collection in more detail.

Future work will focus on one of two aspects: the development of better-performing models for ER trained on this corpus (such as Convolutional Neural Networks, LSTM models, and Transformer-based models e.g. BERT), and the development of models that use the extracted entities from models trained using this corpus as features for tasks such as job recommendation, where

[11] `https://tinyurl.com/skill-extraction-dataset`

candidate CVs are matched with job descriptions that closely match their skill sets.

11. Acknowledgements

This work was supported by the Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation [grant number EP/S023062/1].
We would like to thank TribePad for their support during this research project. We would also like to thank Johann Petrak for his helpful advice regarding inter-annotator agreement statistics.

12. Bibliographical References

Almalis, N. D., Tsihrintzis, G. A., and Karagiannis, N. (2014). A content based approach for recommending personnel for job positions. In *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, pages 45–49. IEEE, jul.

Bastian, M., Hayes, M., Vaughan, W., Shah, S., Skomoroch, P., and Kim, H. (2014). Linked in skills: Large-scale topic extraction and inference. *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, (October):1–8.

Choudhary, S., Koul, S., Mishra, S., Thakur, A., and Jain, R. (2016). Collaborative job prediction based on Naïve Bayes Classifier using python platform. *2016 International Conference on Computation System and Information Technology for Sustainable Solutions, CSITSS 2016*, pages 302–306.

Deleger, L., Li, Q., Lingren, T., Kaiser, M., Molnar, K., Stoutenborough, L., Kouril, M., Marsolo, K., and Solti, I. (2012). Building gold standard corpora for medical natural language processing tasks. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2012:144–153.

F. Tjong Kim Sang, E. and De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé, H., and Crawford, K. (2018). Datasheets for Datasets.

Gugnani, A. and Misra, H. (2020). Implicit Skills Extraction Using Document Embedding and Its Use in Job Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(08):13286–13293, apr.

Hoang, P., Mahoney, T., Javed, F., and McNair, M. (2018). Large-scale occupational skills normalization for online recruitment. *AI Magazine*, 39(1):5–14.

Hovy, D., Berg-Kirkpatrick, T., Vaswani, A., and Hovy, E. (2013). Learning Whom to Trust with MACE. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130.

Hripcsak, G. and Rothschild, A. S. (2005). Agreement, the F-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298.

Karakatsanis, I., AlKhader, W., MacCrory, F., Alibasic, A., Omar, M. A., Aung, Z., and Woon, W. L. (2017). Data mining approach to monitoring the requirements of the job market: A case study. *Information Systems*, 65:1–6, apr.

Khobreh, M., Ansari, F., Fathi, M., Vas, R., Mol, S. T., Berkers, H. A., and Varga, K. (2016). An Ontology-Based Approach for the Semantic Representation of Job Knowledge.

Kivimäki, I., Panchenko, A., Dessy, A., Verdegem, D., Francq, P., Fairon, C., Bersini, H., and Saerens, M. (2020). A graph-based approach to skill extraction from text. *Proceedings of TextGraphs@EMNLP 2013: The 8th Workshop on Graph-Based Methods for Natural Language Processing*, (October):79–87.

Lafferty, J., Mccallum, A., and Pereira, F. (1999). Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data Abstract. 2001(June):282–289.

Lawson, N. and Eustice, K. (2010). Annotating Large Email Datasets for Named Entity Recognition with Mechanical Turk. *Computational Linguistics*, (June):71–79.

Li, J., Sun, A., Han, J., and Li, C. (2020). A Survey on Deep Learning for Named Entity Recognition. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Maheshwari, S., Abhishek, S., and Reddy, P. K. (2010). An Approach to Extract Special Skills to Improve the Performance of Resume Selection. *Conference: Databases in Networked Information Systems, 6th International Workshop, DNIS 2010*, (March 2010).

McInnis, B. and Leshed, G. (2016). Lessons learned: Running user studies with crowd workers. *Interactions*, 23(5):50–53.

Rodrigues, F. and Pereira, F. (2017). Deep learning from crowds.

Sabou, M., Bontcheva, K., Derczynski, L., and Scharl, A. (2014). Corpus annotation through crowdsourcing: Towards best practice guidelines. *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, (2010):859–866.

Shi, B., Yang, J., Guo, F., and He, Q. (2020). Salience and Market-aware Skill Extraction for Job Targeting. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (August):2871–2879.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.