De-identification of Privacy-related Entities in Job Postings

Kristian Nørgaard Jensen,[◇] **Mike Zhang**[◇] **and Barbara Plank**
Department of Computer Science
ITU Copenhagen, Denmark
`krnj@itu.dk, mikz@itu.dk, bplank@itu.dk`

Abstract

De-identification is the task of detecting privacy-related entities in text, such as person names, emails and contact data. It has been well-studied within the medical domain. The need for de-identification technology is increasing, as privacy-preserving data handling is in high demand in many domains. In this paper, we focus on job postings. We present JOB-STACK, a new corpus for de-identification of personal data in job vacancies on Stackoverflow. We introduce baselines, comparing Long-Short Term Memory (LSTM) and Transformer models. To improve upon these baselines, we experiment with contextualized embeddings and distantly related auxiliary data via multi-task learning. Our results show that auxiliary data improves de-identification performance. Surprisingly, vanilla BERT turned out to be more effective than a BERT model trained on other portions of Stackoverflow.

1 Introduction

It is becoming increasingly important to anonymize privacy-related information in text, such as person names and contact details. The task of de-identification is concerned with detecting and anononymizing such information. Traditionally, this problem has been studied in the medical domain by e.g., Szarvas et al. (2007); Friedrich et al. (2019); Trienes et al. (2020) to anonymize (or pseudo-anonymize) person-identifiable information in electronic health records (EHR). With new privacy-regulations (Section 2) de-identification is becoming more important for broader types of text. For example, a company or public institution might seek to

de-identify documents before sharing them. On another line, de-identification can benefit society and technology at scale. Particularly auto-regressive models trained on massive text collections pose a potential risk for exposing private or sensitive information (Carlini et al., 2019, 2020), and de-identification can be one way to address this.

In this paper, we analyze how effectively sequence labeling models are in identifying privacy-related entities in job posts. To the best of our knowledge, we are the first study that investigates de-identification methods applied to job vacancies. In particular, we examine: How do Transformer-based models compare to LSTM-based models on this task (**RQ1**)? How does BERT compare to BERT$_{\text{Overflow}}$ (Tabassum et al., 2020) (**RQ2**)? To what extent can we use existing medical de-identification data and Named Entity Recognition (NER) data to improve de-identification performance (**RQ3**)? To answer these questions, we put forth a new corpus, JOBSTACK, annotated with around 22,000 sentences in English job postings from Stackoverflow for person names, contact details, locations, and information about the profession of the job post itself.

Contributions We present JOBSTACK, the first job postings dataset with professional and personal entity annotations from Stackoverflow. Our experiments on entity de-identification with neural methods show that Transformers outperform bi-LSTMs, but surprisingly a BERT variant trained on another portion of Stackoverflow is less effective. We find auxiliary tasks from both news and the medical domain to help boost performance.

2 Related Work

2.1 De-identification in the Medical Domain

De-identification has mostly been investigated in the medical domain (e.g., Szarvas et al. (2007);

[◇]The authors contributed equally to this work.

Meystre et al. (2010); Liu et al. (2015); Jiang et al. (2017); Friedrich et al. (2019); Trienes et al. (2020)) to ensure the privacy of a patient in the analysis of their medical health records. Apart from an ethical standpoint, it is also a legal requirement imposed by multiple legislations such as the US Health Insurance Portability and Accountability Act (HIPAA) (Act, 1996) and the European General Data Protection Regulation (GDPR) (Regulation, 2016).

Many prior works in the medical domain used the I2B2/UTHealth dataset (Stubbs and Uzuner, 2015) to evaluate de-identification. The dataset consists of clinical narratives, which are free-form medical texts written as a first person account by a clinician. Each of the documents describes a certain event, consultation or hospitalization. All of the texts have been annotated with a set of Protected Health Information (PHI) tags (e.g. name, profession, location, age, date, contact, IDs) and subsequently replaced by realistic surrogates. The dataset was originally developed for use in a shared task for automated de-identification systems. Systems tend to perform very well on this set, in the shared task three out of ten systems achieved F1 scores above 90 (Stubbs et al., 2015). More recently, systems reach over 98 F1 with neural models (Dernoncourt et al., 2017; Liu et al., 2017; Khin et al., 2018; Trienes et al., 2020; Johnson et al., 2020). We took I2B2 as inspiration for annotation of JOBSTACK.

Past methods for de-identification in the medical domain can be categorised in three categories. (1) Rule-based approaches, (2) traditional machine learning (ML)-based systems (e.g., feature-based Conditional Random Fields (CRFs) (Lafferty et al., 2001), ensemble combining CRF and rules, data augmentation, clustering), and (3) neural-based approaches.

Rule-based First, Gupta et al. (2004) made use of a set of rules, dictionaries, and fuzzy string matching to identify protected health information (PHI). In a similar fashion, Neamatullah et al. (2008) used lexical look-up tables, regular expressions, and heuristics to find instances of PHI.

Traditional ML Second, classical ML approaches employ feature-based CRFs (Aberdeen et al., 2010; He et al., 2015). Moreover, earlier work showed the use of CRFs in an ensemble with rules (Stubbs et al., 2015). Other ML approaches

include data augmentation by McMurry et al. (2013), where they added public medical texts to properly distinguish common medical words and phrases from PHI and trained decision trees on the augmented data.

Neural methods Third, regarding neural methods, Dernoncourt et al. (2017) were the first to use Bi-LSTMs, which they used in combination with character-level embeddings. Similarly, Khin et al. (2018) performed de-identification by using a Bi-LSTM-CRF architecture with ELMo embeddings (Peters et al., 2018). Liu et al. (2017) used four individual methods (CRF-based, Bi-LSTM, Bi-LSTM with features, and rule-based methods) for de-identification, and used an ensemble learning method to combine all PHI instances predicted by the three methods. Trienes et al. (2020) opted for a Bi-LSTM-CRF as well, but applied it with contextual string embeddings (Akbik et al., 2018). Most recently, Johnson et al. (2020) fine-tuned $BERT_{base}$ and $BERT_{large}$ (Devlin et al., 2019) for de-identification. Next to "vanilla" BERT, they experiment with fine-tuning different domain specific pre-trained language models, such as SciBERT (Beltagy et al., 2019) and BioBERT (Lee et al., 2020). They achieve state-of-the art performance in de-identification on the I2B2 dataset with the fine-tuned $BERT_{large}$ model. From a different perspective, the approach of Friedrich et al. (2019) is based on adversarial learning, which automatically pseudo-anonymizes EHRs.

2.2 De-identification in other Domains

Data protection in general however is not only limited to the medical domain. Even though work outside the clinical domain is rare, personal and sensitive data is in abundance in all kinds of data. For example, Eder et al. (2019) pseudonymised German emails. Bevendorff et al. (2020) published a large preprocessed email corpus, where only the email addresses themselves where anonymized. Apart from emails, several works went into de-identification of SMS messages (Treurniet et al., 2012; Patel et al., 2013; Chen and Kan, 2013) in Dutch, French, English and Mandarin respectively. Both Treurniet et al. (2012); Chen and Kan (2013) conducted the same strategy and automatically anonymized all occurrences of dates, times, decimal amounts, and numbers with more than one digit (telephone numbers, bank accounts, et cetera), email addresses, URLs, and IP ad-

	Train	Dev	Test	Total
Time	June – August 2020	September 2020		-
# Documents	313	41	41	395
# Sentences	18,055	2082	2092	22,219
# Tokens	195,425	22,049	21,579	239,053
# Entities	4,057	462	426	5,154
avg. # sentences	57.68	50.78	51.02	53.16
avg. tokens / sent.	10.82	10.59	10.32	10.78
avg. entities / sent.	0.22	0.22	0.20	0.21
density	14.73	14.31	14.58	14.54
Organization	1803	215	208	2226
Location	1511	157	142	1810
Profession	558	63	64	685
Contact	99	10	7	116
Name	86	17	5	108

Table 1: Statistics of our JOBSTACK dataset.

dresses. All sensitive information was replaced with a placeholder. Patel et al. (2013) introduced a system to anonymize SMS messages by using dictionaries. It uses a dictionary of first names and anti-dictionaries (of ordinary language and of some forms of SMS writing) to identify the words that require anonymization.

In our work, we study de-identification for names, contact information, addresses, and professions, as further described in Section 3.

3 JOBSTACK Dataset

In this section, we describe the JOBSTACK dataset. There are two basic approaches to remove privacy-bearing data from the job postings. First, anonymization identifies instances of personal data (e.g. names, email addresses, phone numbers) and replaces these strings by some placeholder (e.g. {name}, {email}, {phone}). The second approach, pseudonymisation preserves the information of personal data by replacing these privacy-bearing strings with randomly chosen alternative strings from the same privacy type (e.g. replacing a name with "John Doe"). The term de-identification subsumes both anonymization and pseudonymisation. In this work, we focus on anonymization.[1]

Eder et al. (2019) argues that the anonymization approach might be appropriate to eliminate privacy-bearing data in the medical domain, but would be inappropriate for most Natural Language Processing (NLP) applications since crucial discriminative information and contextual clues will be erased by anonymization.

If we shift towards pseudonymisation, we argue that there is still the possibility to resurface the original personal data. Henceforth, our goal is to anonimize job postings to the extent that one would not be able to easily identify a company from the job posting. However, as job postings are public, we are aware that it would be simple to find the original company that posted it with a search engine. Nevertheless, we abide to the GDPR compliance which requires us to protect the personal data and privacy of EU citizens for transactions that occur within EU member states (Regulation, 2016). In job postings this would be the names of employees, and their corresponding contact information.[2]

Over a period of time, we scraped 2,755 job postings from Stackoverflow and selected 395 documents to annotate, the subset ranges from June 2020 to September 2020. We manually annotated the job postings with the following five entities: Organization, Location, Contact, Name, and Profession.

To make the task as realistic as possible, we kept all sentences in the documents. The statistics pro-

---

[1]Meystre (2015) notes that de-identification means removing or replacing personal identifiers to make it difficult to reestablish a link between the individual and his or her data, but it does not make this link impossible.

[2]https://ec.europa.eu/info/law/
law-topic/data-protection/reform/
rules-business-and-organisations/
application-regulation/
do-data-protection-rules-apply-data-about-company_
en

```
...
13. Job description:
14. [XXX_Organization] is a modern multi tenant, microservices based solution and Floor Planning is one major functional
solution vertical of the [XXX_Organization] platform.

15. What you'll be doing:
16. As a [XXX_Profession] for [XXX_Organization], you will be one of the founding members of our [XXX_Location] based floor planning
development team.
17. You will be in charge for development of future floor planning capabilities on the [XXX_Organization] platform and be the
software architect for the capability.
18. You will drive the team to improve the coding practices and boost performance.
19. You will also be a member of our [XXX_Organization] and have a major influence on feature roadmap and technologies we use.
...
```

Figure 1: Snippet of a job posting, full job posting can be found in Appendix A.

vided in the following therefore reflect the natural distribution of entities in the data. A snippet of an example job post can be seen in Figure 1, the full job posting can be found in Appendix A.

3.1 Statistics

Table 1 shows the statistics of our dataset. We split our data in 80% train, 10% development, and 10% test. Besides of a regular document-level random split, ours is further motivated based on time. The training set covers the job posts posted between June to August 2020 and the development- and test set are posted in September 2020. To split the text into sentences, we use the `sentence-splitter` library used for processing the Europarl corpus (Koehn, 2005). In the training set, we see that the average number of sentences is higher than in the development- and test set (6-7 more). We therefore also calculate the density of the entities, meaning the percentage of sentences with at least one entity. The table shows that 14.5% of the sentences in JOBSTACK contain at least one entity. Note that albeit having document boundaries, we treat the task of de-identification as a standard word-level sequence labeling task.

3.2 Annotation Schema

The aforementioned entity tags are based on the English I2B2/UTHealth corpus (Stubbs and Uzuner, 2015). The tags are more coarse-grained than the I2B2 tags. For example, we do not distinguish between zip code and city, but tag them with `Location`. We give a brief explanation of the tags.

`Organization`: This includes all companies and their legal entity mentioned in the job postings. The tag is not limited to the company that authored the job posting, but does also include men-

tions of stakeholders or any other company.

`Location`: This is the address of the company in the job posting. The location also refers to all other addresses, zip codes, cities, regions, and countries mentioned throughout the text. This is not limited to the company address, but should be used for all location names in the job posting, including abbreviations.

`Contact`: The label includes, URLs, email addresses and phone numbers. This could be, but is not limited to, contact info of an employee from the authoring company.

`Name`: This label covers names of people. This could be, but is not limited to, a person from the company, such as the contact person, CEO, or the manager. All names appearing in the job posting should be annotated no matter the relation to the job posting itself. Titles such as Dr. are not part of the annotation. Apart from people names in our domain, difficulties could arise with other type of names. An example would be project names, with which one could identify a company. In this work, we did not annotate such names.

`Profession`: This label covers the profession that is being searched for in the job posting or desired prior relevant jobs for the current profession. We do not annotate additional meta information such as gender (e.g. Software Engineer (f/m)). We also do not annotate mentions of colleague positions in neither singular or plural form. For example: *"As a Software Engineer, you are going to work with Security Engineers"*. Here we annotate Software Engineer as profession, but we do not annotate Security Engineers. While this may sound straightforward, however, there are difficulties in regards to annotating professions. A job posting is free text, meaning that one can write anything they prefer to make the job posting as clear as possible (e.g., *Software Engineer (at a unicorn start-*

up based in [..]). The opposite is also possible, when they are looking for one applicant to fill in one of multiple positions. For example, "*We are looking for an applicant to fill in the position of DevOps/Software Engineer*". From our interpretation, they either want a "DevOps Engineer" or a "Software Engineer". We decided to annotate the full string of characters "DevOps/Software Engineer" as a profession.

3.3 Annotation Quality

	Token	Entity	Unlabeled
A1 – A2	0.889	0.767	0.892
A1 – A3	0.898	0.782	0.904
A2 – A3	0.917	0.823	0.920
Fleiss' κ	0.902	0.800	0.906

Table 2: Inter-annotator agreement of the annotators. We show agreement over pairs with Cohen's κ and all annotators with Fleiss' κ.

To evaluate our annotation guidelines, a sample of the data was annotated by three annotators, one with a background in Linguistics (A1) and two with a background in Computer Science (A2, A3). We used an open source text annotation tool named `Doccano` (Nakayama et al., 2018). There are around 1,500 overlapping sentences that we calculated agreement on. The annotations were compared using Cohen's κ (Fleiss and Cohen, 1973) between pairs of annotators, and Fleiss' κ (Fleiss, 1971), which generalises Cohen's κ to more than two concurrent annotations. Table 2 shows three levels of κ calculations, we follow Balasuriya et al. (2009)'s approach of calculating agreement in NER. (1) `Token` is calculated on the token level, comparing the agreement of annotators on each token (including non-entities) in the annotated dataset. (2) `Entity` is calculated on the agreement between named entities alone, excluding agreement in cases where all annotators agreed that a token was not a named-entity. (3) `Unlabeled` refers to the agreement between annotators on the exact span match over the surface string, regardless of the type of named entity (i.e., we only check the position of tag without regarding the type of the named entity). Landis and Koch (1977) state that a κ value greater than 0.81 indicates almost perfect agreement. Given this, all annotators are in strong agreement.

After this annotation quality estimation, we finalized the guidelines. They formed the basis for the professional linguist annotator, who annotated and finalized the entire final JOBSTACK dataset.

4 Methods

For entity de-identification we use a classic Named Entity Recognition (NER) approach using a Bi-LSTM with a CRF layer. On top of this we evaluate the performance of Transformer-based models with two different pre-trained BERT variants. Furthermore, we evaluate the helpfulness of auxiliary tasks, both using data close to our domain, such as de-identification of medical notes, and more general NER, which covers only a subset of the entities. Further details on the data are given in Section 4.3.

4.1 Models

Firstly, we test a Bi-LSTM sequence tagger (Bilty) (Plank et al., 2016), both with and without a CRF layer. The architecture is similar to the widely used models in previous works. For example, preliminary results of Bilty versus Trienes et al. (2020) show accuracy almost identical to each other: 99.62% versus 99.76%. Next we test a Transformer based model, namely the MaChAmp (van der Goot et al., 2021) toolkit. Current research shows good results for NER using a Transformer model without a CRF layer (Martin et al., 2020), hence we tested MaChAmp both with and without a CRF layer for predictions. For both models, we use their default parameters.

4.2 Embeddings

For embeddings, we tested with no pre-trained embeddings, pre-trained Glove (Pennington et al., 2014) embeddings, and Transformer-based pre-trained embeddings. For Transformer-based embeddings we focused our attention on two BERT models, $BERT_{base}$ (Devlin et al., 2019) and $BERT_{Overflow}$ (Tabassum et al., 2020). When using the Transformer-based embeddings with the Bi-LSTM, the embeddings were fixed and did not get updated during training.

Using the MaChAmp (van der Goot et al., 2021) toolkit, we fine-tune the BERT variant with a Transformer encoder. For the Bi-LSTM sequence tagger, we first derive BERT representations as input to the tagger. The tagger further uses word

Model	F1 Score	Precision	Recall
Bilty	71.76 ± 2.57	79.00 ± 1.10	65.80 ± 3.72
Bilty + CRF	75.15 ± 0.66	84.09 ± 1.90	67.96 ± 0.81
Bilty + Glove 50d	72.53 ± 0.83	79.21 ± 2.19	67.03 ± 2.76
Bilty + Glove 50d + CRF	72.74 ± 2.23	82.93 ± 0.87	64.93 ± 3.93
Bilty + BERT$_{base}$	77.99 ± 0.91	83.70 ± 0.58	73.01 ± 1.34
Bilty + BERT$_{base}$ + CRF	80.09 ± 0.60	$\mathbf{88.23 \pm 0.87}$	73.30 ± 1.47
Bilty + BERT$_{Overflow}$	52.01 ± 3.15	70.86 ± 0.68	41.27 ± 4.19
Bilty + BERT$_{Overflow}$ + CRF	53.08 ± 2.88	77.79 ± 1.20	40.33 ± 2.98
MaChAmp + BERT$_{base}$	85.70 ± 0.13	86.66 ± 0.73	84.78 ± 0.44
MaChAmp + BERT$_{base}$ + CRF	$\mathbf{86.27 \pm 0.31}$	86.40 ± 0.62	$\mathbf{86.15 \pm 0.00}$
MaChAmp + BERT$_{Overflow}$	65.84 ± 0.48	70.88 ± 0.17	61.47 ± 0.81
MaChAmp + BERT$_{Overflow}$ + CRF	69.35 ± 0.96	77.27 ± 3.68	63.06 ± 2.11

Table 3: Results on the development set across three runs using our JOBSTACK dataset.

Model	Auxiliary tasks	F1 Score	Precision	Recall
Bilty + BERT$_{base}$ + CRF	JOBSTACK + CoNLL	81.90 ± 0.32	86.91 ± 1.94	77.49 ± 1.87
	JOBSTACK + I2B2	79.15 ± 2.19	83.61 ± 2.61	75.18 ± 2.59
	JOBSTACK + CoNLL + I2B2	81.37 ± 2.01	84.92 ± 1.67	78.28 ± 4.34
Bilty + BERT$_{Overflow}$ + CRF	JOBSTACK + CoNLL	58.62 ± 1.46	79.34 ± 2.34	46.54 ± 1.99
	JOBSTACK + I2B2	55.99 ± 1.93	72.03 ± 6.48	46.10 ± 2.55
	JOBSTACK + CoNLL + I2B2	59.15 ± 2.15	71.20 ± 4.80	50.86 ± 3.31
MaChAmp + BERT$_{base}$ + CRF	JOBSTACK + CoNLL	$\mathbf{87.20 \pm 0.34}$	87.24 ± 1.94	$\mathbf{87.23 \pm 1.24}$
	JOBSTACK + I2B2	86.64 ± 0.53	$\mathbf{88.44 \pm 0.84}$	84.92 ± 0.44
	JOBSTACK + CoNLL + I2B2	86.06 ± 0.66	86.13 ± 0.50	86.00 ± 0.87
MaChAmp + BERT$_{Overflow}$ + CRF	JOBSTACK + CoNLL	70.62 ± 0.64	75.65 ± 1.41	66.24 ± 0.98
	JOBSTACK + I2B2	73.88 ± 0.16	80.26 ± 1.32	68.47 ± 1.03
	JOBSTACK + CoNLL + I2B2	73.29 ± 0.22	77.66 ± 0.82	69.41 ± 0.89

Table 4: Performance of multi-task learning on the development set across three runs.

and character embeddings which are updated during model training.

The BERT$_{Overflow}$ model is a transformer with the same architecture as BERT$_{base}$. It has been trained from scratch on a large corpus of text from the Q&A section of Stackoverflow, making it closer to our text domain than the "vanilla" BERT model. However, BERT$_{Overflow}$ is not trained on the job postings portion of Stackoverflow.

4.3 Auxiliary tasks

Both the Bi-LSTM (Plank et al., 2016) and the MaChAmp (van der Goot et al., 2021) toolkit are capable of Multi Task Learning (MTL) (Caruana, 1997). We therefore, set up a number of experiments testing the impact of three different auxiliary tasks. The auxiliary tasks and their datasets are as follows:

- I2B2/UTHealth (Stubbs and Uzuner, 2015) - Medical de-identification;

- CoNLL 2003 (Sang and De Meulder, 2003) - News Named Entity Recognition;

- The combination of the above.

The data of the two tasks are similar to our dataset in two different ways. The I2B2 lies in a different text domain, namely medical notes, however, the label set of the task is close to our label set, as mentioned in Section 3.2. For CoNLL, we have a general corpus of named entities but fewer types (location, organization, person, and miscellaneous), but the text domain is presumably closer to our data. We test the impact of using both auxiliary tasks along with our own dataset.

Model	Auxiliary tasks	F1 Score	Precision	Recall
Bilty + $\text{BERT}_{\text{base}}$ + CRF	JOBSTACK	78.99 ± 0.32	$\mathbf{82.44 \pm 0.95}$	75.90 ± 1.39
MaChAmp + $\text{BERT}_{\text{base}}$ + CRF	JOBSTACK	79.91 ± 0.38	75.92 ± 0.39	84.35 ± 0.49
	JOBSTACK + CoNLL	81.27 ± 0.28	77.84 ± 1.19	85.06 ± 0.91
	JOBSTACK + I2B2	$\mathbf{82.05 \pm 0.80}$	80.30 ± 0.99	83.88 ± 0.67
	JOBSTACK + CoNLL + I2B2	81.47 ± 0.43	77.66 ± 0.58	$\mathbf{85.68 \pm 0.57}$

Table 5: Evaluation of the best performing models on the test set across three runs.

5 Evaluation

Here we will outline the results of the experiments described in Section 4. All results are mean scores across three different runs.[3] The metrics are all calculated using the `conlleval` script[4] from the original CoNLL-2000 shared task. Table 3 shows the results from training on JOBSTACK only, Table 4 shows the results of the MTL experiments described in Section 4.3. Both report results on the development set. Lastly, Table 5 shows the scores from evaluating selected best models as found on the development set, when tested on the final held-out test set.

Is a CRF layer necessary? In Table 3, as expected, adding the CRF for the Bi-LSTM clearly helps, and consistently improves precision and thereby F1 score. For the stronger BERT model the overall improvement is smaller and does not necessarily stem from higher precision. We note that on average across the three seed runs, MaChAmp with $\text{BERT}_{\text{base}}$ and no CRF mistakenly adds an I-tag following an O-tag 8 times out of 426 gold entities. In contrast, the MaChAmp with $\text{BERT}_{\text{base}}$ and CRF, makes no such mistake in any of its three seed runs. Earlier research, such as Souza et al. (2019) show that BERT models with a CRF layer improve or perform similarly to its simpler variants when comparing the overall F1 scores. Similarly, they note that in most cases it shows higher precision scores but lower recall, as in our results for the development set. However, interestingly, the precision drops during test for the Transformer-based model. As the overall F1 score increases slightly, we use the CRF layer in all subsequent experiments. The main take-away here is that both models benefit from an added CRF layer for the task, but the Transformer model

to a smaller degree.

LSTM versus Transformer Initially, LSTM networks dominated the de-identification field in the medical domain. Up until recently, large-scale pre-trained language models have been ubiquitous in NLP, although rarely used in this field. On both development and test results (Table 3, Table 5), we show that a Transformer-based model outperforms the LSTM-based approaches with non-contextualized and contextualized representations.

Poor performance with $\text{BERT}_{\text{Overflow}}$ $\text{BERT}_{\text{base}}$ is the best embedding method among all experiments using Bilty, with $\text{BERT}_{\text{Overflow}}$ being the worst with a considerable margin. Being able to fine-tune $\text{BERT}_{\text{base}}$ does give a good increase in performance overall. The same trend is apparent with fine-tuning $\text{BERT}_{\text{Overflow}}$, but it is not enough to catch up with $\text{BERT}_{\text{base}}$. We see that overall MaChAmp with $\text{BERT}_{\text{base}}$ and CRF is the best model. However, Bilty with $\text{BERT}_{\text{base}}$ and CRF does have the best precision.

We hypothesized the domain-specific $\text{BERT}_{\text{Overflow}}$ representations would be beneficial for this task. Intuitively, $\text{BERT}_{\text{Overflow}}$ would help with detecting profession entities. Profession entities contain specific skills related to the IT domain, such as *Python developer*, *Rust developer*, *Scrum master*. Although the corpus it is trained on is not one-to-one to our vacancy domain, we expected to see at most a slight performance drop. This is not the case, as the drop in performance turned out to be high. It is not fully clear to us why this is the case. It could be the Q&A data it is trained on consists of more informal dialogue than in job postings. In the future, we would like to compare these results to training a BERT model on job postings data.

Auxiliary data increases performance Looking at the results from the auxiliary experiments in Table 4 we see that all auxiliary sources are ben-

eficial, for both types of models. A closer look reveals that once again MaChAmp with BERT$_{base}$ is the best performer across all three auxiliary tasks. Also, we see that Bilty with BERT$_{base}$ has good precision, though not the best this time around. For a task like de-identification recall is preferable, thereby showing that fine-tuning BERT is better than the classic Bi-LSTM-CRF. Moreover, we see that BERT$_{Overflow}$ is under-performing compared to BERT$_{base}$. However, BERT$_{Overflow}$ is able to get a four point increase in F1 with I2B2 as auxiliary task in MaChAmp. For Bilty with BERT$_{Overflow}$ we see a slightly greater gain with both CoNLL and I2B2 as auxiliary tasks. When comparing the auxiliary data sources to each other, we note that the closer text domain (CoNLL news) is more beneficial than the closer label set (I2B2) from a more distant medical text source. This is consistent for the strongest models.

In general, it can be challenging to train multi-task networks that outperform or even match their single-task counterparts (Alonso and Plank, 2017; Clark et al., 2019). Ruder (2017) mentions training on a large number of tasks is known to help regularize multi-task models. A related benefit of MTL is the transfer of learned "knowledge" between closely related tasks. In our case, it has been beneficial to add auxiliary tasks to improve our performance on both development and test compared to a single task setting. In particular, it seemed to have helped with pertaining a high recall score.

Performance on the test set   Finally, we evaluate the best performing models on our held out test set. The best models are selected based on their performance on F1, precision, and recall. The results are seen in Table 5. Comparing the results to those seen in Table 3 and Table 4 it is clear to see that Bilty with BERT$_{base}$ sees a smaller drop in F1 compared to that of MaChAmp with BERT$_{base}$. We do also see an increase in recall for Bilty compared to its performance on the development set. In general we see that recall for each model is staying quite stable without any significant drops. It is also interesting to see that, the internal ranking between MTL MaChAmp with BERT$_{base}$ has changed, with JOBSTACK + I2B2 being the best performing model in terms of F1.

Per-entity Analysis   In Table 6, we show a deeper analysis on the test set: the performance of the two different auxiliary tasks in a multi-

| Entity | | MaChAmp + JOBSTACK | |
		+ CoNLL	+ I2B2
Organization (208)	F1	77.51 ± 0.81	78.34 ± 1.32
	P	73.73 ± 1.66	77.86 ± 1.60
	R	81.73 ± 0.96	78.85 ± 1.74
Location (142)	F1	86.88 ± 1.51	86.67 ± 1.80
	P	83.86 ± 1.82	83.47 ± 1.19
	R	90.14 ± 1.41	90.14 ± 2.54
Profession (64)	F1	80.20 ± 2.76	83.88 ± 0.90
	P	77.44 ± 3.82	82.42 ± 0.63
	R	83.33 ± 4.51	85.42 ± 1.80
Contact (7)	F1	87.91 ± 3.81	75.48 ± 4.30
	P	90.47 ± 8.25	71.03 ± 4.18
	R	85.71 ± 0.00	80.95 ± 8.24
Name (5)	F1	86.25 ± 8.08	85.86 ± 4.38
	P	76.39 ± 12.03	75.40 ± 6.87
	R	100.00 ± 0.00	100.00 ± 0.00

Table 6: Performance of the two different auxiliary tasks. Reported is the F1, Precision (P), and Recall (R) per entity. The number behind the entity name is the gold label instances in the test set.

task learning setting, namely CoNLL and I2B2. We hypothesized different performance gains with each auxiliary task. For I2B2, we expected `Contact` and `Profession` to do better than CoNLL, since I2B2 contains contact information entities (e.g., phone numbers, emails, et cetera) and professions of patients. Surprisingly, this is not the case for `Contact`, as CoNLL outperforms I2B2 on all three metrics. We do note however this result could be due to little instances of `Contact` and `Name` being present in the gold test set. Additionally, both named entities are predicted six to nine times by both models on all three runs on the test set. This could indicate the strong difference in performance. For `Profession`, it shows that I2B2 is beneficial for this particular named entity as expected. For the other three named entities, the performance is similar. As `Location`, `Name`, and `Organization` are in both datasets, we did not expect any difference in performance. The results confirm this intuition.

6   Conclusions

In this work we introduce JOBSTACK, a dataset for de-identification of English Stackoverflow job postings. Our implementation is publicly available.[5] The dataset is freely available upon request.
  We present neural baselines based on LSTM

---

and Transformer models. Our experiments show the following: (1) Transformer-based models consistently outperform Bi-LSTM-CRF-based models that have been standard for de-identification in the medical domain (**RQ1**). (2) Stackoverflow-related BERT representations are not more effective than regular BERT representations on Stackoverflow job postings for de-identification (**RQ2**). (3) MTL experiments with BERT representations and related auxiliary data sources improve our de-identification results (**RQ3**); the auxiliary task trained on the closer text type was the most beneficial, yet results improved with both auxiliary data sources. This shows the benefit of using multi-task learning for de-identification in job vacancy data.

Acknowledgements

References

John Aberdeen, Samuel Bayer, Reyyan Yeniterzi, Ben Wellner, Cheryl Clark, David Hanauer, Bradley Malin, and Lynette Hirschman. 2010. The mitre identification scrubber toolkit: design, training, and assessment. *International journal of medical informatics*, 79(12):849–859.

Accountability Act. 1996. Health insurance portability and accountability act of 1996. *Public law*, 104:191.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 44–53.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R Curran. 2009. Named entity recognition in wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, pages 10–18.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611.

Janek Bevendorff, Khalid Al Khatib, Martin Potthast, and Benno Stein. 2020. Crawling and preprocessing mailing lists at scale for dialog analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1151–1158.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 267–284.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2020. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Tao Chen and Min-Yen Kan. 2013. Creating a live, public short message service corpus: the nus sms corpus. *Language Resources and Evaluation*, 47(2):299–335.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc V Le. 2019. Bam! born-again multi-task networks for natural language understanding. *arXiv preprint arXiv:1907.04829*.

Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2017. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Elisabeth Eder, Ulrike Krieg-Holz, and Udo Hahn. 2019. De-identification of emails: Pseudonymizing privacy-sensitive data in a german email corpus. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 259–269.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

Max Friedrich, Arne Köhn, Gregor Wiedemann, and Chris Biemann. 2019. Adversarial learning of privacy-preserving text representations for de-identification of medical records. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5829–5839, Florence, Italy. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive Choice, Ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the Software Demonstrations of the 16th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Dilip Gupta, Melissa Saul, and John Gilbertson. 2004. Evaluation of a deidentification (de-id) software engine to share pathology reports and clinical documents for research. *American journal of clinical pathology*, 121(2):176–186.

Bin He, Yi Guan, Jianyi Cheng, Keting Cen, and Wenlan Hua. 2015. Crfs based de-identification of medical records. *Journal of biomedical informatics*, 58:S39–S46.

Zhipeng Jiang, Chao Zhao, Bin He, Yi Guan, and Jingchi Jiang. 2017. De-identification of medical records using conditional random fields and long short-term memory networks. *Journal of biomedical informatics*, 75:S43–S53.

Alistair EW Johnson, Lucas Bulgarelli, and Tom J Pollard. 2020. Deidentification of free-text medical records using pre-trained bidirectional transformers. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 214–221.

Kaung Khin, Philipp Burckhardt, and Rema Padman. 2018. A deep learning architecture for de-identification of patient notes: Implementation and evaluation. *arXiv preprint arXiv:1810.01570*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

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

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Zengjian Liu, Yangxin Chen, Buzhou Tang, Xiaolong Wang, Qingcai Chen, Haodi Li, Jingfeng Wang, Qiwen Deng, and Suisong Zhu. 2015. Automatic de-identification of electronic medical records using token-level and character-level conditional random fields. *Journal of biomedical informatics*, 58:S47–S52.

Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. 2017. De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of biomedical informatics*, 75:S34–S42.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Andrew J McMurry, Britt Fitch, Guergana Savova, Isaac S Kohane, and Ben Y Reis. 2013. Improved de-identification of physician notes through integrative modeling of both public and private medical text. *BMC medical informatics and decision making*, 13(1):112.

Stephane M Meystre. 2015. De-identification of unstructured clinical data for patient privacy protection. In *Medical Data Privacy Handbook*, pages 697–716. Springer.

Stephane M Meystre, F Jeffrey Friedlin, Brett R South, Shuying Shen, and Matthew H Samore. 2010. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology*, 10(1):70.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from https://github.com/doccano/doccano.

Ishna Neamatullah, Margaret M Douglass, H Lehman Li-wei, Andrew Reisner, Mauricio Villarroel, William J Long, Peter Szolovits, George B Moody, Roger G Mark, and Gari D Clifford. 2008. Automated de-identification of free-text medical records. *BMC medical informatics and decision making*, 8(1):32.

Namrata Patel, Pierre Accorsi, Diana Inkpen, Cédric Lopez, and Mathieu Roche. 2013. Approaches of anonymisation of an sms corpus. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 77–88. Springer.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany. Association for Computational Linguistics.

General Data Protection Regulation. 2016. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. *Official Journal of the European Union (OJ)*, 59(1-88):294.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*.

Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. 2015. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *Journal of biomedical informatics*, 58:S11–S19.

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of Biomedical Informatics*, 58:S20–S29.

György Szarvas, Richárd Farkas, and Róbert Busa-Fekete. 2007. State-of-the-art anonymization of medical records using an iterative machine learning framework. *Journal of the American Medical Informatics Association*, 14(5):574–580.

Jeniya Tabassum, Mounica Maddela, Wei Xu, and Alan Ritter. 2020. Code and named entity recognition in stackoverflow. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Maaske Treurniet, Orphee De Clercq, Henk van den Heuvel, and Nelleke Oostdijk. 2012. Collection of a corpus of dutch sms. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2268–2273.

J Trienes, D Trieschnigg, C Seifert, and D Hiemstra. 2020. Comparing rule-based, feature-based and deep neural methods for de-identification of dutch medical records. In *Eickhoff, C.(ed.), Health Search and Data Mining Workshop: Proceedings of the ACM WSDM 2020 Health Search and Data Mining Workshop co-located with the 13th ACM International WSDM Conference (WSDM 2020) Houston, Texas, USA, February 3, 2020*, pages 3–11. [Sl]: CEUR.

A Example Job Posting

1. [**XXX**_{Profession}]

2. [**XXX**_{Organization}]

3. <ADDRESS>, <ADDRESS>, [**XXX**_{Location}] , - , [**XXX**_{Location}]

4. Date posted: 2020−08−13
5. Likes: 0, Dislikes: 0, Love: 0
6. Salary: SALARY
7. Job type: FULL_TIME
8. Experience level: Mid–Level, Senior, Lead
9. Industry: Big Data, Cloud-Based Solutions, Enterprise Software
10. Company size: 501−1
11. Company type: Private

12. Technologies: c#, typescript, cad, 2d, 3d

13. Job description:
14. [**XXX**_{Organization}] is a modern multi tenant, microservices based solution and Floor Planning is one major functional solution vertical of the [**XXX**_{Organization}] platform.

15. What you'll be doing:
16. As a [**XXX**_{Profession}] for [**XXX**_{Organization}], you will be one of the founding members of our [**XXX**_{Location}] based floor planning development team.
17. You will be in charge for development of future floor planning capabilities on the [**XXX**_{Organization}] platform and be the software architect for the capability.
18. You will drive the team to improve the coding practices and boost performance.
19. You will also be a member of our [**XXX**_{Organization}] and have a major influence on feature roadmap and technologies we use.

20. What you'll bring to the table:
21. Solid software design and development skills and at least 5 year experience in the industry
22. Good understanding of CAD type of software in 2D and 3D worlds
23. Experience on rendering technologies, 2D/3D data models and data types
24. Hands-on experience in implementing CAD designers / drafting / drawing tools for on-line use
25. C#, C++, TypeScript or Angular/React knowledge
26. Strong ambition to deliver great quality software and to continuously improve the way we do development
27. Good spoken and written English
28. Ability to work on-site in our [**XXX**_{Location}] office, with flexible remote work possibilities

29. What we consider as an advantage:
30. Eagerness to find out and learn about the latest computer graphics technologies, and also to share your findings
31. Knowledge of OpenDesign components (Teigha)

32. What we offer you in return:
33. An international career and learning opportunities in a rapidly growing software company
34. A fun, ambitious, and committed team of smart people to work with
35. A respectful and professional, yet easy-going atmosphere where individual thinking is encouraged
36. Responsibilities in challenging projects from day one
37. A position where you can help retailers fight against food waste

38. Are you the one we're looking for?
39. Apply today and become a part of our [**XXX**_{Organization}] family!
40. You can apply by sending your cover letter and resume through the application form as soon as possible, but no later than 31st of August.
41. Please note that we will fill this position as soon as we've found the right person, so we recommend that you act quickly.
42. If you have questions, [**XXX**_{Name}] ([**XXX**_{Contact}]) from our Recruitment team is happy to answer them.
43. Also kindly note that we cannot process any applications through email.

44. Job benefits:

53. Company description:
54. [**XXX**_{Organization}] is a fast-growing software company developing products that help retail companies plan and operate more efficiently.
55. By accurately forecasting consumption of goods, we reduce inventory costs, increase availability and cut waste.
56. Helping retailers eliminate food spoilage and reduce fleet emissions from transportation has a significant environmental impact as well!