De-identification of Privacy-related Entities in Job Postings

Kristian Nørgaard Jensen
krnj@itu.dk

Mike Zhang
mikz@itu.dk

Barbara Plank
bapl@itu.dk

nlpnorth.github.io

IT UNIVERSITY OF COPENHAGEN

De-identification

What is de-identification?

Remove entities that can identify persons or companies*, to make the re-identification of such entities harder. To comply to the GDPR (2016) regulations.

Before: Founded by Brandon Beck and Marc Merrill, and currently under the leadership of CEO Nicolo Laurent, we're headquartered in Los Angeles, California

After: Founded by [XXX_{Name}] and [XXX_{Name}], and currently under the leadership of CEO [XXX_{Name}], we're headquartered in [$XXX_{Location}$]

*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

Motivation

Motivation

- Mostly applied in the medical domain (Stubbs and Uzuner, 2015)
  - De-identification of Electronic Health Records
  - Personal data not only limited to this domain!

- Use de-identification on job-postings
  - Remove person/company names, contact info, professions, addresses

Before: European Bioinformatics Institute (EMBL - EBI) - Wellcome Trust Genome Campus, CB10 1SA, Hinxton, -, GB

After: [XXX$_{\text{Organization}}$]([XXX$_{\text{Organization}}$]) - [XXX$_{\text{Location}}$]

Research Questions

Research Questions

1. **How do Transformer-based models compare to LSTM-based models on this task?**
   a. Bi-LSTMs (Graves et al., 2005) have shown to work well for de-identification (Trienes et al., 2020) how does a transformer-based model fare?

2. **How does $\text{BERT}_{\text{base}}$ compare to a domain specific BERT ($\text{BERT}_{\text{Overflow}}$)?**
   a. Would a domain specific pre-trained BERT perform better than $\text{BERT}_{\text{base}}$?

3. **To what extent can we use auxiliary data to improve de-identification performance?**
   a. A related benefit of MTL (Caruana, 1997) is the transfer of learned “knowledge” between closely related tasks, which then helps improve performance.

Experimental Setup

JobStack

	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

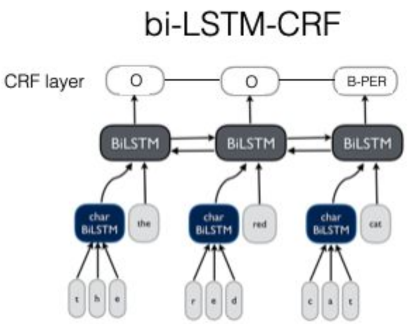
- Job postings from Stackoverflow;
- Time-based data split;
- **Annotating** *Organization*, *Location*, *Profession*, *Contact*, and *Name*;
- 3 annotators.

	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

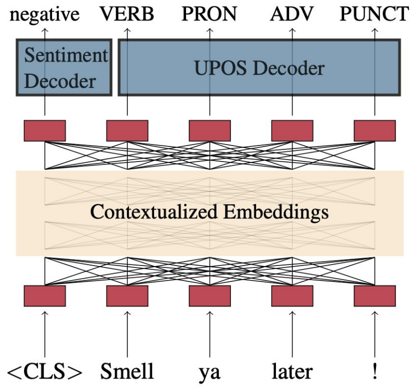
Models

- Bi-LSTM sequence tagger (*Bilty)*
  - with(out) CRF layer

- Transformer based model (MaChAmp)
  - with(out) CRF layer
  - **BERT**$_{\textbf{base}}$ (Devlin et al., 2019)
  - **BERT**$_{\textbf{overflow}}$(Tabassum et al., 2020)
    - BERT$_{\text{base}}$ architecture;
    - Q&A section of Stackoverflow.



Bilty
(Plank et al., 2016)



MaChAmp
(van der Goot et al., 2021)

RQ1: Transformer vs. Bi-LSTM
RQ2: $\text{BERT}_{\text{base}}$ vs. $\text{BERT}_{\text{Overflow}}$

Results on dev

Model	F1 Score	Precision	Recall
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

- **Bilty** vs. **MaChAmp**
  - High F1 and recall with transformer-based model;
  - High Precision with Bi-LSTM model.
- **BERT**$_{\textbf{base}}$ performs better than **BERT**$_{\textbf{Overflow}}$
- **CRF**-layer helps with performance

RQ3: Auxiliary Data

Results on dev

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

- Both models are capable of Multi-Task Learning (MTL; Caruana, 1997)
- **Two** auxiliary tasks:
  1. **CoNLL** (Sang et al., 2003)
  2. **I2B2** (Stubbs et al., 2015)
- **Transformer-based** model performs best.

Results on test

Model	Auxiliary tasks	F1 Score	Precision	Recall
Bilty + BERT$_{base}$ + CRF	JobStack	78.99 ± 0.32	$\mathbf{82.44 \pm 0.95}$	75.90 ± 1.39
MaChAmp + BERT$_{base}$ + 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	$\mathbf{85.68 \pm 0.57}$

- Best performing models on dev applied to test;
- Similar to dev:
  - High F1 and recall with transformer-based model;
  - High Precision with Bi-LSTM model.
- Auxiliary data **helps** improving de-identification performance.
- **Do we need a CRF layer?**
  MaChAmp with BERT$_{base}$ without a CRF layer adds an I-tag following an O-tag 8 times out of 426 gold entities.

Per-entity Analysis on test

Entity		MaChAmp + + CoNLL	MaChAmp + + 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

- Specific auxiliary task would give different performance gains:
  - **I2B2**: Contact and Profession
  - **CoNLL**: Location, Organization, Name

- I2B2 beneficial for Profession as expected, not with contact.
- CoNLL not as impactful as expected.

Conclusions

Conclusions

- Introduced a new dataset: JobStack
- **RQ1**: Transformer vs. Bi-LSTM
  - Transformer models outperform Bi-LSTM models
- **RQ2**: $\text{BERT}_{\text{base}}$ vs. $\text{BERT}_{\text{Overflow}}$
  - Domain specific $\text{BERT}_{\text{Overflow}}$ is outperformed by $\text{BERT}_{\text{base}}$
- **RQ3**: MTL
  - Using auxiliary data helps improve de-identification performance

Thank you!



Kristian Nørgaard Jensen
krnj@itu.dk

Mike Zhang
mikz@itu.dk

Barbara Plank
bapl@itu.dk

github.com/kris927b/JobStack
nlpnorth.github.io

IT UNIVERSITY OF COPENHAGEN

References

[1] Caruana, R. (1997). Multitask learning. *Machine learning*, *28*(1), 41-75.

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). 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).

[3] van der Goot, R., Üstün, A., Ramponi, A., Sharaf, I., & Plank, B. (2020). Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. *arXiv preprint arXiv:2005.14672*.

[4] Alex Graves and J ̈urgen Schmidhuber. 2005. Frame[wise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks, 18(5):602–610.

[5] Plank, B., Søgaard, A., & Goldberg, Y. (2016, August). 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)* (pp. 412-418).

[6] Sang, E. T. K., & De Meulder, F. (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* (pp. 142-147).

[7] Stubbs, A., & Uzuner, Ö. (2015). Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of biomedical informatics*, *58*, S20-S29.

[8] Tabassum, J., Maddela, M., Xu, W., & Ritter, A. (2020, July). Code and Named Entity Recognition in StackOverflow. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[9] Trienes, J., Trieschnigg, D., Seifert, C., & Hiemstra, D. (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* (pp. 3-11). [Sl]: CEUR.