Leveraging Subword Embeddings for Multinational Address Parsing [1]

Marouane Yassine, David Beauchemin,
François Laviolette, Luc Lamontagne

Département d'informatique et de génie logiciel,
Université Laval

*marouane.yassine.1@ulaval.ca, david.beauchemin.5@ulaval.ca,
francois.laviolette@ift.ulaval.ca, luc.lamontagne@ift.ulaval.ca*

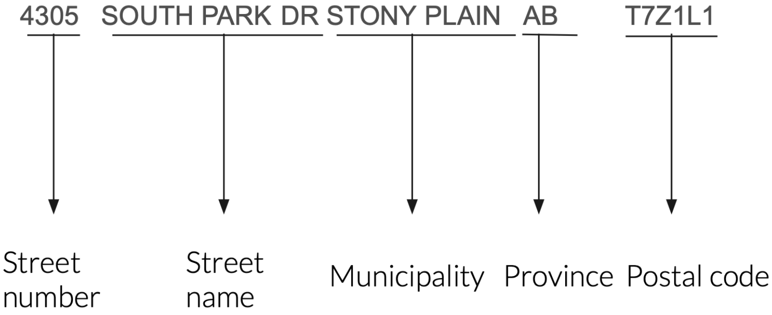
July 14 2020

1. https://arxiv.org/abs/2006.16152

Agenda

Introduction

What's address parsing ?



Useful for tasks such as *Record Linkage* and *Geocoding*.

Agenda

Related Work - Address Parsing for One Country

- Rule based methods

Related Work - Address Parsing for One Country

- Rule based methods
- Probabilistic models
  - Hidden Markov Models (HMM) [Li et al., 2014]
  - Conditional Random Fields (CRF) [Wang et al., 2016]

Related Work - Address Parsing for One Country

- Rule based methods
- Probabilistic models
  - Hidden Markov Models (HMM) [Li et al., 2014]
  - Conditional Random Fields (CRF) [Wang et al., 2016]
- Neural networks
  - Feed-forward Neural Network [Sharma et al., 2018]
  - Recurrent Neural Networks [Mokhtari et al., 2019]

Related work - Multinational address parsing

Libpostal [1]

- CRF based model

1. https://github.com/openvenues/libpostal
2. https://medium.com/@albarrentine/statistical-nlp-on-openstreetmap-part-2-80405b988718

Related work - Multinational address parsing

Libpostal [1]

- CRF based model
- Preprocessing and postprocessing

1. https://github.com/openvenues/libpostal
2. https://medium.com/@albarrentine/statistical-nlp-on-openstreetmap-part-2-80405b988718

Related work - Multinational address parsing

Libpostal [1]

- CRF based model
- Preprocessing and postprocessing
- '*[T]rained on over 1 billion examples in every inhabited country on Earth*' [2]

1. https://github.com/openvenues/libpostal
2. https://medium.com/@albarrentine/statistical-nlp-on-openstreetmap-part-2-80405b988718

Related work - Multinational address parsing

Libpostal [1]

- CRF based model
- Preprocessing and postprocessing
- '*[T]rained on over 1 billion examples in every inhabited country on Earth*' [2]

No previous neural network approaches for multinational address parsing

1. https://github.com/openvenues/libpostal
2. https://medium.com/@albarrentine/statistical-nlp-on-openstreetmap-part-2-80405b988718

Agenda

Subword Embeddings

Word embedding : vector representation of a word

- Non-contextual embeddings (e.g : *Word2Vec*, *Glove*)
- Contextual embeddings (e.g : *ELMo*, *BERT*)

Subword Embeddings

Word embedding : vector representation of a word

- Non-contextual embeddings (e.g : *Word2Vec*, *Glove*)
- Contextual embeddings (e.g : *ELMo*, *BERT*)

Subword embedding : vector representation of a unit

- Character level
- Character *n-grams* (e.g : *fastText*)
- Byte pair embeddings (*BPEmb*) [Heinzerling and Strube, 2017]

Why subword embeddings ?

Multilingual setting

- Need of alignement vectors (*Muse* [Conneau et al., 2017])
- *fasText* support for OOV
- *MultiBPEmb* pre-trained on 275 languages

Agenda

Embedding Model

We compare two pre-trained embedding.

Embedding Model

We compare two pre-trained embedding.

- A fixed pre-trained monolingual fastText model (pre-trained on the French language) (**fastText**).

Embedding Model

We compare two pre-trained embedding.

- A fixed pre-trained monolingual fastText model (pre-trained on the French language) (**fastText**).
- A encoding of words using MultiBPEmb and merge the obtained embeddings for each word into one word embedding using a Bidirectional LSTM (Bi-LSTM) (hidden state dimension of 300). We refer to this embeddings model technique as **BPEmb**.

Embedding Model

We compare two pre-trained embedding.

- A fixed pre-trained monolingual fastText model (pre-trained on the French language) (**fastText**).
- A encoding of words using MultiBPEmb and merge the obtained embeddings for each word into one word embedding using a Bidirectional LSTM (Bi-LSTM) (hidden state dimension of 300). We refer to this embeddings model technique as **BPEmb**.

We run a comparison of the two methods (**fastText** and **BPEmb**) to evaluate which one gives better results in our setting.

Tagging Model

We use a Seq2Seq model consisting of

Tagging Model

We use a Seq2Seq model consisting of

- a one-layer unidirectional LSTM encoder

Tagging Model

We use a Seq2Seq model consisting of

- a one-layer unidirectional LSTM encoder
- a one-layer unidirectional LSTM decoder

Tagging Model

We use a Seq2Seq model consisting of

- a one-layer unidirectional LSTM encoder
- a one-layer unidirectional LSTM decoder
- a fully-connected linear layer with a softmax activation.

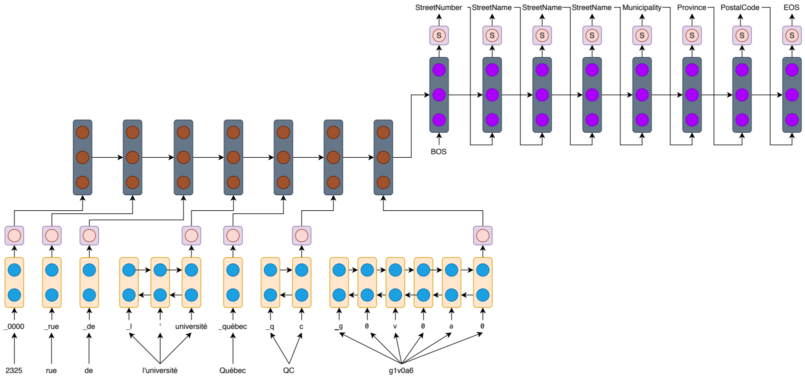
Tagging Model

We use a Seq2Seq model consisting of

- a one-layer unidirectional LSTM encoder
- a one-layer unidirectional LSTM decoder
- a fully-connected linear layer with a softmax activation.

Both the encoder's and decoder's hidden states are of dimension 1024.

Architecture



About the Data

- Built using the open-source data on which Libpostal's models were trained.

1. Libpostal used 20 tags.

About the Data

- Built using the open-source data on which Libpostal's models were trained.
- Contain 61 countries.
- We used eight tags : StreetNumber, StreetName, Unit, Municipality, Province, PostalCode, Orientation, and GeneralDelivery [1].

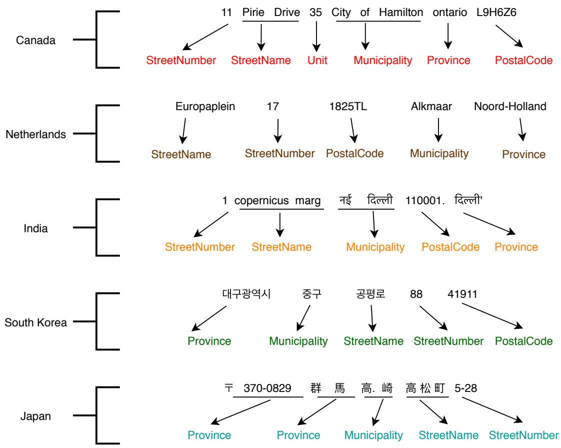
1. Libpostal used 20 tags.

Examples of Address and Their Patterns

We use five different address patterns (one for each color) and another for some countries' using more than a pattern (no color).

Examples of Address and Their Patterns

We use five different address patterns (one for each color) and another for some countries' using more than a pattern (no color).



Training Data and Holdout Test Set

20 countries are used for multinational training with a sample size of $100,000$ per country. The rest is used as a holdout (table below).

Country	Number of samples	Country	Number of samples	Country	Number of samples	Country	Number of samples
United States	8,000,000	Germany	1,576,059	Poland	459,522	Czechia	195,269
Brazil	8,000,000	Spain	1,395,758	Norway	405,649	Italy	178,848
South Korea	6,048,106	Netherlands	1,202,173	Austria	335,800	France	20,050
Australia	5,428,043	Canada	910,891	Finland	280,219	United Kingdom	14,338
Mexico	4,853,349	Switzerland	474,240	Denmark	199,694	Russia	8115

Zero-Shot Test Set

41 countries are used for zero-shot transfer evaluation (never seen in training) (table below).

Country	Number of samples	Country	Number of samples	Country	Number of samples	Country	Number of samples
Belgium	66,182	Slovenia	9773	Réunion	2514	Singapore	968
Sweden	32,291	Ukraine	9554	Moldova	2376	Bangladesh	888
Argentina	27,692	Belarus	7590	Indonesia	2259	Paraguay	839
India	26,084	Serbia	6792	Bermuda	2065	Cyprus	836
Romania	19,420	Croatia	5671	Malaysia	2043	Bosnia	681
Slovakia	18,975	Greece	4974	South Africa	1388	Ireland	638
Hungary	17,460	New Zealand	4678	Latvia	1325	Algeria	601
Japan	14,089	Portugal	4637	Kazakhstan	1087	Colombia	569
Venezuela	10,696	Lithuania	3126	New Caledonia	1036	Uzbekistan	505
Philippines	10,471	Faroe Islands	2982	Estonia	1024		

Agenda

Training Procedure

- We trained five times [1] each of our two model (**fastText** and **BPEmb**).

1. With the following seed $\{5, 10, 15, 20, 25\}$.

Training Procedure

- We trained five times [1] each of our two model (**fastText** and **BPEmb**).
- They were trained during 200 epochs with a batch size of 2048.

1. With the following seed $\{5, 10, 15, 20, 25\}$.

Training Procedure

- We trained five times[1] each of our two model (**fastText** and **BPEmb**).
- They were trained during 200 epochs with a batch size of 2048.
- We used an early stopping with a patience of 15 epochs.

1. With the following seed $\{5, 10, 15, 20, 25\}$.

Training Procedure

- We trained five times[1] each of our two model (**fastText** and **BPEmb**).
- They were trained during 200 epochs with a batch size of 2048.
- We used an early stopping with a patience of 15 epochs.
- A starting learning rate at 0.1 and a learning rate scheduling (factor of 0.1) after ten epochs without loss decrease.

1. With the following seed $\{5, 10, 15, 20, 25\}$.

Training Procedure

- We trained five times[1] each of our two model (**fastText** and **BPEmb**).
- They were trained during 200 epochs with a batch size of 2048.
- We used an early stopping with a patience of 15 epochs.
- A starting learning rate at 0.1 and a learning rate scheduling (factor of 0.1) after ten epochs without loss decrease.
- Cross-Entropy loss.

1. With the following seed $\{5, 10, 15, 20, 25\}$.

Training Procedure

- We trained five times[1] each of our two model (**fastText** and **BPEmb**).
- They were trained during 200 epochs with a batch size of 2048.
- We used an early stopping with a patience of 15 epochs.
- A starting learning rate at 0.1 and a learning rate scheduling (factor of 0.1) after ten epochs without loss decrease.
- Cross-Entropy loss.
- Stochastic Gradient Descent (SGD) optimizer.

1. With the following seed $\{5, 10, 15, 20, 25\}$.

Training Procedure

- We trained five times [1] each of our two model (**fastText** and **BPEmb**).
- They were trained during 200 epochs with a batch size of 2048.
- We used an early stopping with a patience of 15 epochs.
- A starting learning rate at 0.1 and a learning rate scheduling (factor of 0.1) after ten epochs without loss decrease.
- Cross-Entropy loss.
- Stochastic Gradient Descent (SGD) optimizer.
- Teacher forcing [Williams and Zipser, 1989].

1. With the following seed $\{5, 10, 15, 20, 25\}$.

Training Procedure

- We trained five times[1] each of our two model (**fastText** and **BPEmb**).
- They were trained during 200 epochs with a batch size of 2048.
- We used an early stopping with a patience of 15 epochs.
- A starting learning rate at 0.1 and a learning rate scheduling (factor of 0.1) after ten epochs without loss decrease.
- Cross-Entropy loss.
- Stochastic Gradient Descent (SGD) optimizer.
- Teacher forcing [Williams and Zipser, 1989].
- Trained using Poutyne [Paradis, 2018].

1. With the following seed $\{5, 10, 15, 20, 25\}$.

Multinational Evaluation

Country	**FastText**	**BPEmb**	Country	**FastText**	**BPEmb**
United States	$\mathbf{99.61 \pm 0.09}$	98.55 ± 2.19	Poland	$\mathbf{99.69 \pm 0.07}$	99.19 ± 1.39
Brazil	$\mathbf{99.40 \pm 0.10}$	98.54 ± 1.68	Norway	$\mathbf{99.46 \pm 0.06}$	97.98 ± 1.31
South Korea	99.96 ± 0.01	$\mathbf{99.99 \pm 0.02}$	Austria	$\mathbf{99.28 \pm 0.03}$	98.28 ± 1.56
Australia	$\mathbf{99.68 \pm 0.05}$	99.21 ± 1.17	Finland	$\mathbf{99.77 \pm 0.03}$	99.72 ± 0.30
Mexico	$\mathbf{99.60 \pm 0.06}$	98.55 ± 2.22	Denmark	$\mathbf{99.71 \pm 0.07}$	99.20 ± 1.38
Germany	$\mathbf{99.77 \pm 0.04}$	99.23 ± 1.30	Czechia	$\mathbf{99.57 \pm 0.09}$	98.77 ± 2.22
Spain	$\mathbf{99.75 \pm 0.05}$	98.65 ± 2.36	Italy	$\mathbf{99.73 \pm 0.05}$	98.91 ± 1.76
Netherlands	$\mathbf{99.61 \pm 0.07}$	99.26 ± 1.23	France	$\mathbf{99.66 \pm 0.08}$	98.65 ± 2.00
Canada	$\mathbf{99.79 \pm 0.05}$	99.19 ± 1.33	United Kingdom	$\mathbf{99.61 \pm 0.10}$	98.66 ± 2.11
Switzerland	$\mathbf{99.53 \pm 0.09}$	99.49 ± 0.53	Russia	$\mathbf{99.03 \pm 0.24}$	97.52 ± 4.23

Multinational Evaluation

Country	**FastText**	**BPEmb**	Country	**FastText**	**BPEmb**
United States	$\mathbf{99.61 \pm 0.09}$	98.55 ± 2.19	Poland	$\mathbf{99.69 \pm 0.07}$	99.19 ± 1.39
Brazil	$\mathbf{99.40 \pm 0.10}$	98.54 ± 1.68	Norway	$\mathbf{99.46 \pm 0.06}$	97.98 ± 1.31
South Korea	99.96 ± 0.01	$\mathbf{99.99 \pm 0.02}$	Austria	$\mathbf{99.28 \pm 0.03}$	98.28 ± 1.56
Australia	$\mathbf{99.68 \pm 0.05}$	99.21 ± 1.17	Finland	$\mathbf{99.77 \pm 0.03}$	99.72 ± 0.30
Mexico	$\mathbf{99.60 \pm 0.06}$	98.55 ± 2.22	Denmark	$\mathbf{99.71 \pm 0.07}$	99.20 ± 1.38
Germany	$\mathbf{99.77 \pm 0.04}$	99.23 ± 1.30	Czechia	$\mathbf{99.57 \pm 0.09}$	98.77 ± 2.22
Spain	$\mathbf{99.75 \pm 0.05}$	98.65 ± 2.36	Italy	$\mathbf{99.73 \pm 0.05}$	98.91 ± 1.76
Netherlands	$\mathbf{99.61 \pm 0.07}$	99.26 ± 1.23	France	$\mathbf{99.66 \pm 0.08}$	98.65 ± 2.00
Canada	$\mathbf{99.79 \pm 0.05}$	99.19 ± 1.33	United Kingdom	$\mathbf{99.61 \pm 0.10}$	98.66 ± 2.11
Switzerland	$\mathbf{99.53 \pm 0.09}$	99.49 ± 0.53	Russia	$\mathbf{99.03 \pm 0.24}$	97.52 ± 4.23

- **FastText** gives the best performance across the board without considering the standard deviation.

Multinational Evaluation

Country	**FastText**	**BPEmb**	Country	**FastText**	**BPEmb**
United States	$\mathbf{99.61 \pm 0.09}$	98.55 ± 2.19	Poland	$\mathbf{99.69 \pm 0.07}$	99.19 ± 1.39
Brazil	$\mathbf{99.40 \pm 0.10}$	98.54 ± 1.68	Norway	$\mathbf{99.46 \pm 0.06}$	97.98 ± 1.31
South Korea	99.96 ± 0.01	$\mathbf{99.99 \pm 0.02}$	Austria	$\mathbf{99.28 \pm 0.03}$	98.28 ± 1.56
Australia	$\mathbf{99.68 \pm 0.05}$	99.21 ± 1.17	Finland	$\mathbf{99.77 \pm 0.03}$	99.72 ± 0.30
Mexico	$\mathbf{99.60 \pm 0.06}$	98.55 ± 2.22	Denmark	$\mathbf{99.71 \pm 0.07}$	99.20 ± 1.38
Germany	$\mathbf{99.77 \pm 0.04}$	99.23 ± 1.30	Czechia	$\mathbf{99.57 \pm 0.09}$	98.77 ± 2.22
Spain	$\mathbf{99.75 \pm 0.05}$	98.65 ± 2.36	Italy	$\mathbf{99.73 \pm 0.05}$	98.91 ± 1.76
Netherlands	$\mathbf{99.61 \pm 0.07}$	99.26 ± 1.23	France	$\mathbf{99.66 \pm 0.08}$	98.65 ± 2.00
Canada	$\mathbf{99.79 \pm 0.05}$	99.19 ± 1.33	United Kingdom	$\mathbf{99.61 \pm 0.10}$	98.66 ± 2.11
Switzerland	$\mathbf{99.53 \pm 0.09}$	99.49 ± 0.53	Russia	$\mathbf{99.03 \pm 0.24}$	97.52 ± 4.23

- **FastText** gives the best performance across the board without considering the standard deviation.
- South Korean results are excellent despite the completely different alphabet.

Multinational Evaluation

- When using standard deviation, the **BPEmb** model achieves better results than **fastText** in most cases.

Multinational Evaluation

- When using standard deviation, the **BPEmb** model achieves better results than **fastText** in most cases.
- We find that South Korea is the only country where a perfect accuracy ($100\ \%$) was achieved using **BPEmb** (3 out of 5).

Multinational Evaluation

- When using standard deviation, the **BPEmb** model achieves better results than **fastText** in most cases.
- We find that South Korea is the only country where a perfect accuracy (100 %) was achieved using **BPEmb** (3 out of 5).
- Randomly reordering 6000 South Korean address as either the first (red) or the second (brown) address pattern (equally divided between the two), the mean accuracy drops to 28.04% (the mean accuracy is of 12.29 % using a random tags procedure).

Zero-shot Evaluation

Country	FastText	BPEmb	Country	FastText	BPEmb
Belgium	**88.14 ± 1.04**	87.45 ± 1.37	Faroe Islands	74.14 ± 1.83	**86.59 ± 2.21**
Sweden	81.59 ± 4.53	**88.30 ± 2.92**	Réunion	**96.80 ± 0.45**	92.42 ± 2.38
Argentina	**86.26 ± 0.47**	86.00 ± 4.40	Moldova	**90.18 ± 0.79**	78.11 ± 16.79
India	69.09 ± 1.74	**76.33 ± 7.77**	Indonesia	64.31 ± 0.84	**69.25 ± 2.81**
Romania	**94.49 ± 1.52**	90.52 ± 2.35	Bermuda	92.31 ± 0.60	**92.65 ± 1.84**
Slovakia	82.10 ± 0.98	**89.40 ± 5.09**	Malaysia	78.93 ± 3.78	**92.76 ± 2.55**
Hungary	**48.92 ± 3.59**	24.61 ± 3.35	South Africa	**95.31 ± 1.68**	92.75 ± 7.43
Japan	**41.41 ± 3.21**	33.34 ± 3.83	Latvia	**93.66 ± 0.64**	72.46 ± 5.77
Iceland	96.55 ± 1.20	**97.61 ± 0.98**	Kazakhstan	86.33 ± 3.06	**88.28 ± 11.32**
Venezuala	**94.87 ± 0.53**	89.82 ± 5.74	New Caledonia	**99.48 ± 0.15**	96.44 ± 5.64
Philippines	77.76 ± 3.97	**78.00 ± 11.75**	Estonia	**87.08 ± 1.89**	76.18 ± 1.62
Slovenia	95.37 ± 0.23	**96.47 ± 2.05**	Singapore	**86.42 ± 2.36**	83.23 ± 6.38
Ukraine	**92.99 ± 0.70**	90.86 ± 2.90	Bangladesh	78.61 ± 0.43	**79.77 ± 3.65**
Belarus	**91.08 ± 3.08**	90.16 ± 11.89	Paraguay	96.01 ± 1.23	**96.22 ± 1.78**
Serbia	**95.31 ± 0.48**	88.49 ± 7.05	Cyprus	**97.67 ± 0.34**	92.92 ± 6.94
Croatia	**94.59 ± 2.21**	88.17 ± 4.58	Bosnia	**84.04 ± 1.47**	80.53 ± 6.56
Greece	**81.98 ± 0.60**	35.30 ± 13.51	Ireland	**87.44 ± 0.69**	84.93 ± 2.85
New Zealand	94.27 ± 1.50	**97.77 ± 3.23**	Algeria	**85.37 ± 2.05**	79.66 ± 11.68
Portugal	**93.65 ± 0.46**	90.13 ± 4.47	Colombia	**87.81 ± 0.92**	87.60 ± 3.61
Bulgaria	**91.03 ± 2.07**	87.44 ± 11.94	Uzbekistan	**86.76 ± 1.13**	73.75 ± 3.42
Lithuania	**87.67 ± 3.05**	75.67 ± 2.19			

Zero-shot Evaluation

- 50 % (19 out of 41) near state-of-the-art performance (> 90 %) for **fastText**. Or 35 % for **BPEmb**.

Zero-shot Evaluation

- $50\ \%$ (19 out of 41) near state-of-the-art performance ($> 90\ \%$) for **fastText**. Or $35\ \%$ for **BPEmb**.
- $80\ \%$ (34 out of 41) good performance ($> 80\ \%$) for **fastText**. Or $65\ \%$ for **BPEmb**.

Zero-shot Evaluation

- 50 % (19 out of 41) near state-of-the-art performance (> 90 %) for **fastText**. Or 35 % for **BPEmb**.
- 80 % (34 out of 41) good performance (> 80 %) for **fastText**. Or 65 % for **BPEmb**.
- The lowest results (below 70%) occur for countries where the address pattern and the country official language were not seen in the training data such as India, Hungary, and Japan.

Zero-shot Evaluation

For Hungary and Japan, the poorest results of all are mostly due to the address structure (blue), which is the near inverse of the two most present ones (red and brown) (never seen structure and language).

But, Kazakhstan, which uses the same address pattern as Japan, achieves better results. The main difference being the presence of one of the official language (Kazakh and **Russian**) in the training dataset.

Agenda

Future Work

- Attention mechanism [Bahdanau et al., 2014].

Future Work

- Attention mechanism [Bahdanau et al., 2014].
- Domain-adversarial training techniques (e. g. DANN [Ganin et al., 2015], ADANN [Côté-Allard et al., 2020]) .

Conclusion

- Tackled the multinational address parsing problem with SOTA results.

Conclusion

- Tackled the multinational address parsing problem with SOTA results.
- Showed that subword embeddings help to solve the multilingual aspect of our task.

Conclusion

- Tackled the multinational address parsing problem with SOTA results.
- Showed that subword embeddings help to solve the multilingual aspect of our task.
- We explored the possibility of zero-shot transfer across countries and achieved interesting, but not yet optimal results.

Conclusion

- Tackled the multinational address parsing problem with SOTA results.
- Showed that subword embeddings help to solve the multilingual aspect of our task.
- We explored the possibility of zero-shot transfer across countries and achieved interesting, but not yet optimal results.

For more, read the full article `https://arxiv.org/abs/2006.16152`

Acknowledgment

- This research was supported by the Natural Sciences and Engineering Research Council of Canada (IRCPJ 529529-17) and a Canadian insurance company.
- François Laviolette and Luc Lamontagne for their mentorship in this research.
- Our colleagues at the GRAAL for their comments and reviews.

References I

Bahdanau, D., Cho, K., and Bengio, Y. (2014).
Neural Machine Translation by Jointly Learning to Align and Translate.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017).
Word Translation Without Parallel Data.

Côté-Allard, U., Campbell, E., Phinyomark, A., Laviolette, F., Gosselin, B., and Scheme, E. (2020).
Interpreting deep learning features for myoelectric control : A comparison with handcrafted features.
Frontiers in Bioengineering and Biotechnology, 8 :158.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2015).
Domain-adversarial training of neural networks.

References II

Heinzerling, B. and Strube, M. (2017).
BPEmb : Tokenization-free Pre-trained Subword Embeddings in 275 Languages.

Li, X., Kardes, H., Wang, X., and Sun, A. (2014).
HMM-Based Address Parsing : Efficiently Parsing Billions of Addresses on MapReduce.
In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 433–436. Association for Computing Machinery.

Mokhtari, S., Mahmoody, A., Yankov, D., and Xie, N. (2019).
Tagging Address Queries in Maps Search.
Proceedings of the AAAI Conference on Artificial Intelligence, 33 :9547–9551.

References III

Paradis, F. (2018).
Poutyne : A Keras-like framework for PyTorch.
https://poutyne.org.

Sharma, S., Ratti, R., Arora, I., Solanki, A., and Bhatt, G. (2018).
Automated Parsing of Geographical Addresses : A Multilayer Feedforward Neural Network Based Approach.
In *IEEE 12th International Conference on Semantic Computing*, pages 123–130.

References IV

Wang, M., Haberland, V., Yeo, A., Martin, A., Howroyd, J., and Bishop, J. M. (2016).
A Probabilistic Address Parser Using Conditional Random Fields and Stochastic Regular Grammar.
In *16th International Conference on Data Mining Workshops*, pages 225–232.

Williams, R. J. and Zipser, D. (1989).
A Learning Algorithm for Continually Running Fully Recurrent Neural Networks.
Neural Computation, 1(2) :270–280.