ADRIAAN LEMMENS

NLP Engineer

Find your metric for success and optimize for it.

📍 Leuven 3001, Belgium / @ adriaan.jm.lemmens@gmail.com / ☎ +32 472 69 03 80 / 1992
lemontheme / in adriaan-lemmens

ABOUT ME

I'm a linguist-turned-programmer with over **two years** of experience applying **Natural Language Processing** (NLP) methods to a wide array of real-world challenges in **diverse domains**. I like building production-ready solutions that combine **machine learning**, old-school tricks, and a healthy dose of **pragmatism**. I <3 **Python**, but also enjoy the exercise of learning new programming languages. In my free time, when I'm not reading up on the field, or carrying out toy experiments, I'm usually working on pet projects or making modest open-source contributions.

EXPERIENCE

2017 – Aug. 2019 **Computational Linguist ⟶ NLP Engineer** **NLP Town**
Developed tailor-made solutions (software & models) for clients in various industries/domains, such as E-commerce, Translation, GovTech, Media, Health, Education, and Legal. On average, projects ranged from one to eight months in duration, sometimes involving on-site consulting on my part. In January 2019, I was promoted to NLP Engineer to reflect the fact that my responsibilities had evolved to cover most levels of the development stack.

Spring 2017 **MAI SLT Intern** **NLP Town**
Prototyped a content-driven recommender engine for eBooks, using genre, emotion, and topic assignment predictions + clever document segmentation strategies.

EDUCATION

2016 – 2017 **MSc. Artificial Intelligence (MAI)** **KU Leuven**
Program: Speech & Language Technology (SLT).
Relevant subjects: Natural Language Processing, Machine Learning, Speech Recognition, Text-based Information Retrieval, Language Engineering Applications.
Internship: see EXPERIENCE.

2015 – 2016 **MA Linguistics** **KU Leuven**
Program: Formal & Computational Linguistics.
Thesis title: Pictograph-to-text translation with *Depicto*, a formal grammar-based translation system for use in assistive writing applications.

2011 – 2015 **BA Languages & Literature** **KU Leuven**
Specialization: Synchronic linguistics.
Discovered the joy of programming thanks to an introductory elective on Computational Linguistics. Began teaching myself R, Perl, and Python, and taking an interest in statistical methods.

2006 – 2010 **IGCSEs & International Baccalaureate** **British International School of Shanghai, Pudong**

SKILLS

Python – Upwards of 30K SLOC*, split across exploratory notebooks, data and training pipelines, pet projects, distributed web services, and many things in between; up-to-date with new language features and with the state of the library ecosystem; comfortable writing tests as well as profiling and debugging; huge fan of type annotations; familiar with available tools for optimized numeric computation; currently learning Cython. (∗ ← 'source lines of code' is a questionable metric, but it gives an indication.)

Natural Language Processing – Experienced building NLP-based services for use in real-world applications. At one point or other, I have implemented tailor-made and/or benchmarked off-the-shelf solutions to most well-known NLP tasks, such as classification, sequence labeling, word/document representation learning, information extraction, parsing, extractive summarization, text simplification, machine translation, automatic post-editing, topic modeling, document similarity, clustering, keyword extraction, question answering, coreference resolution, and conversational agents. I'm well-versed in bootstrapping data for domain-specific learning tasks. Most of the projects I've worked on involved a combination of English, Dutch, French, and German.

Machine Learning – Working knowledge of major learning algorithms, including their inductive biases. Practical experience with both classic ('shallow') methods and deep learning; however, due to data and project time constraints in current role, I've spent more time employing the former. Very productive with scikit-learn API, but also familiar with Keras API. Two of my pet interests are reproducibility and model deployment.

Search & Analytics – Retrieving relevant information from large document collections is one of my favorite applications of NLP. For use cases with strict runtime & stability requirements, my go-to tool is the Elasticsearch engine. For more experimental use cases, where flexibility trumps performance guarantees, I've also built my own search engines. One of my goals for 2019 is to explore the possibility of using PyLucene as a means to bridge the performance gap with the Lucene–Solr–Elasticsearch trinity while allowing for richer, semantic-based document representations and custom relevance functions implemented in pure Python.

Software development & architecture – I've designed numerous NLP applications from the ground up, applying Lean principles in order to deliver quickly and to deal efficiently with changing requirements. In terms of architectural style, I enjoy borrowing ideas from the higher-level paradigms of both OOP and Functional Programming. With respect to NLP in particular, I'm always on the lookout for elegant abstractions, regulary drawing inspiration from the source code of open-source libraries.

Linguistic intuition – Thanks to formal training in linguistic analysis, I'm not afraid to get creative when it comes to feature engineering, forming hypotheses about model architectures, adapting pipelines to new languages, and evaluating datasets for possible biases that do not hold in the domain where trained models will ultimately be used.

Other

- **Containerization** with Docker – Going on two years of enthusiastic usage. Aware of best practices (and pitfalls) for writing Dockerfiles and deploying microservices.
- **Mentoring** – Onboarding developers at client companies; hosting workshops; helping interns get started and reviewing their code; 1-on-1 Python tutoring.
- **Other langs** (From 'proficient' to 'I've read at least one book') – Regex, 'the Unix toolkit' (bash, awk, sed, grep, make, etc.), Java, SQL, Make, Clojure, Go, C, Prolog, Perl, R, Elm, Nim.

OPEN SOURCE

- Improved **spaCy**'s preprocessing functionality for Dutch on multiple fronts, including lemmatization.
- On the roadmoap: Open-sourcing **trefwurd** & **wurdbits** – Python libraries for lemmatization and byte-pair encoding, respectively, which grew out of an intellectual exercise to to improve spaCy's lemmatization without resorting to statistical/neural methods.