Adding models for new languages to the spaCy toolkit

[Toolkit extensions]

Intro

- What is spaCy
- Features and advantages
- Adding a new language to the Spacy toolkit.
- What are the alpha support and pre-trained model?
- Steps to pre-trained model.

Body

What is spaCy

spaCy is a free, open-source library for advanced national language processing framework in Python, which helps the developers to build real-world production-ready applications. It is open-source, has a strong community, and currently supports 64+ languages, and is one of the modern frameworks for the NLP. According to the official site spacy.io, it is ready to build actual products, blazing-fast, and has a great ecosystem.

Features

It supports multi-languages, and right now, it does support 64+ world languages. Here are some features from the official site:

Tokenization	Segmenting text into words, punctuations mark, etc.
Part-of-speech (POS) Tagging	Assigning word types to tokens, like verb or noun.
Dependency Parsing	Assigning syntactic dependency labels, describing the relations between individual tokens, like subject or object.
Lemmatization	Assigning the base forms of words. For example, the lemma of “was” is “be”, and the lemma of “rats” is “rat”.
Sentence Boundary Detection (SBD)	Finding and segmenting individual sentences.
Named Entity Recognition (NER)	Labelling named “real-world” objects, like persons, companies or locations.
Entity Linking (EL)	Disambiguating textual entities to unique identifiers in a knowledge base.
Similarity	Comparing words, text spans and documents and how similar they are to each other.
Text Classification	Assigning categories or labels to a whole document, or parts of a document.
Rule-based Matching	Finding sequences of tokens based on their texts and linguistic annotations, similar to regular expressions.
Training	Updating and improving a statistical model’s predictions.
Serialization	Saving objects to files or byte strings.

The main advantage is the spaCy processes the document as objects and builds words as word vectors! Another essential advantage of spaCy is, it is highly customizable. It has transform-based pipelines, a new training system, and project templates, and more. These are allowed to add languages to the spaCy framework nicely!

Adding a new language to the Spacy framework

To add a new language to the spaCy, developers needed to contribute to the spaCy source. It is not easy and complex steps are required, which you will need to follow the rules and conventions of the community. Here is the link to how to contribute:
https://github.com/explosion/spaCy/blob/master/CONTRIBUTING.md

It would be needed to add basic tokenization features and some more necessary rules as written as python files into the corresponding language folder. Also, you will need to implement test cases first as well. The beauty of spaCy is that developers can also build and plug pre-trained language models into the spaCy. It is the next step of adding a new language to the spaCy.

What are the alpha support and pre-trained model?

In the spaCy, languages initially needed to be added as marked “alpha support” model. Alpha support languages usually only include tokenization rules and various other rules and language data. Below is a list of features that can implement as an alpha support model.

- Examples. Sample sentences to test spaCy and its language models.
- Stop words. Stop words for the language.
- Syntax iterators. Detect base noun phrases from a dependency parse. Works on both Doc and Span.
- Lemmatizer. Some rules of assigning the base forms of words.
- Lexical attributes. Names for the numbers to detect numbers.
- Punctuations. Punctuation rules that can separate sentences.
- Tokenizer exceptions. Some exception rules for the tokenizer.

Also, you will need to write the test cases for the new language that is wanting to add.

Steps to pre-trained model.

One of the main advantages of spaCy is it can install and use language models as pre-trained pipelines. Pre-trained language models are the next step of the NLP. It helps to understand better the text information, not only the basic tokenization.

According to the spaCy community website, to go from alpha support to a pre-trained model, the process requires the following steps and components:

- Language data: shipped with spaCy, basically all languages including those marked as “alpha support”. The tokenization should be reliable, and there should be a tag map that maps the tags used in the training data to coarse-grained tags like NOUN and optional morphological features.
- Training corpus: the model needs to be trained on a suitable corpus, e.g. an existing Universal Dependencies treebank. Commercial-friendly treebank licenses are always a plus. Data for tagging and parsing is usually easier to find than data for named entity recognition – in the long term, we want to do more data annotation ourselves using Prodigy, but that's obviously a much bigger project. In the meantime, we have to use other available resources (academic etc.).
- Data conversion: spaCy comes with a range of built-in converters via the spacy convert command that take .conllu files and output spaCy's JSON format. Example of a training pipeline with data conversion is: https://spacy.io/usage/training#spacy-train-cli. Corpora can have very subtle formatting differences, so it's important to check that they can be converted correctly.
- Training pipeline: if we have language data plus a suitable training corpus plus a conversion pipeline, we can run spacy train to train a new model.

Conclusion

The spaCy toolkit is a powerful and highly customizable NLP framework nowadays. It has a strong community, countless features and keeps growing. The NLP is a complex and challenging task, and spaCy is helping to get it to do better. Adding a new language to spaCy is a multiple-step process, yet it will give great features for developing and understanding other languages because of the significant spaCy ecosystems. Each language has its own very different structures, and the spaCy provides precisely those features which each language is developing its processing methods. It is an excellent feature for all languages!

References:

Official site:	https://spacy.io/
spaCy 101:	https://spacy.io/usage/spacy-101
Trained Models & Pipelines:	https://spacy.io/models
Models & Languages:	https://spacy.io/usage/models
spaCy source on the github:	https://github.com/explosion/spaCy
Adding models for new languages:	https://github.com/explosion/spaCy/discussions/3056
Contributing:	https://github.com/explosion/spaCy/blob/master/CONTRIBUTING.md
spaCy tests:	https://github.com/explosion/spaCy/blob/master/spacy/tests/README.md