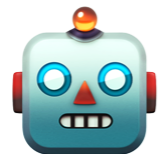# A neural coreference system for conversational agents

HuggingFace Inc.

🤗

Coreference resolution ?

- Linking together mentions that relates to real world entities

Coreference resolution ?

- Linking together mentions that relates to real world entities

Coreference resolution ?

- Linking together mentions that relates to real world entities

My sister has a friend called John
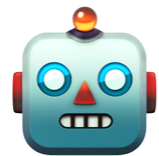
Really, tell me more about him

She thinks he is so funny 🤣

Coreference resolution ?

- Linking together mentions that relates to real world entities
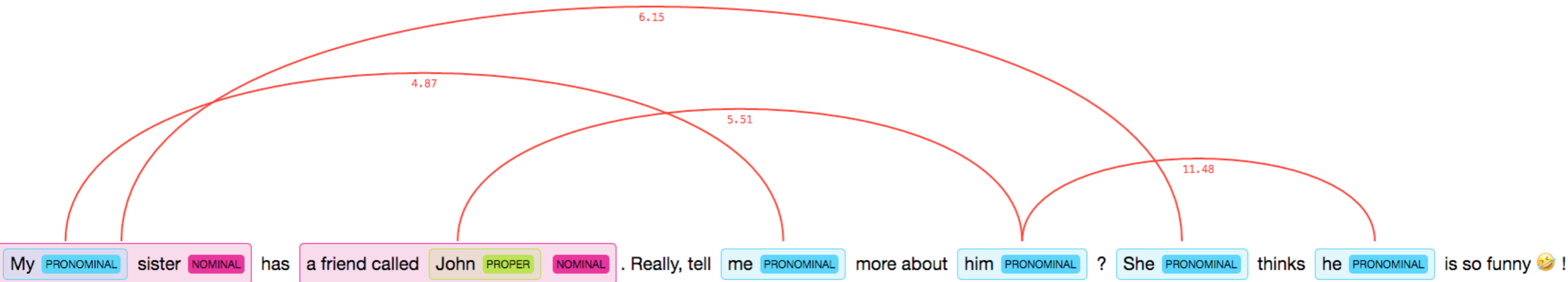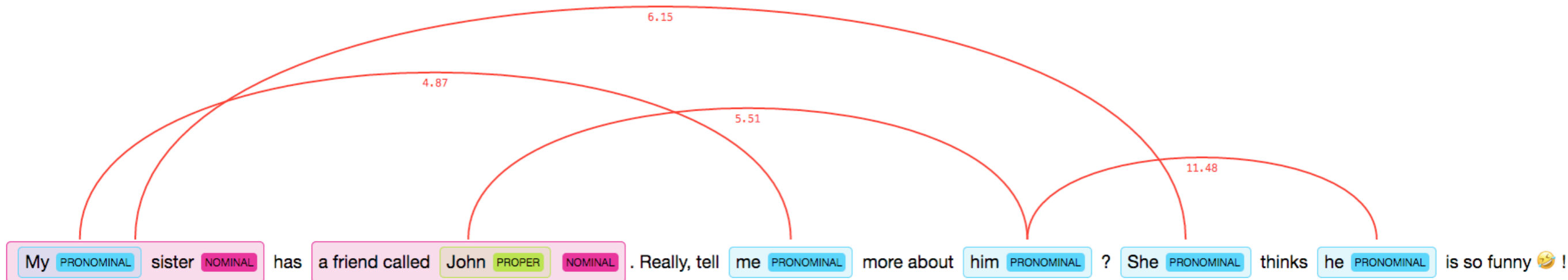
Coreference resolution ?

- Linking together mentions that relates to real world entities

Coreference resolution ?

- Linking together mentions that relates to real world entities

- Try it at: https://huggingface.co/coref/
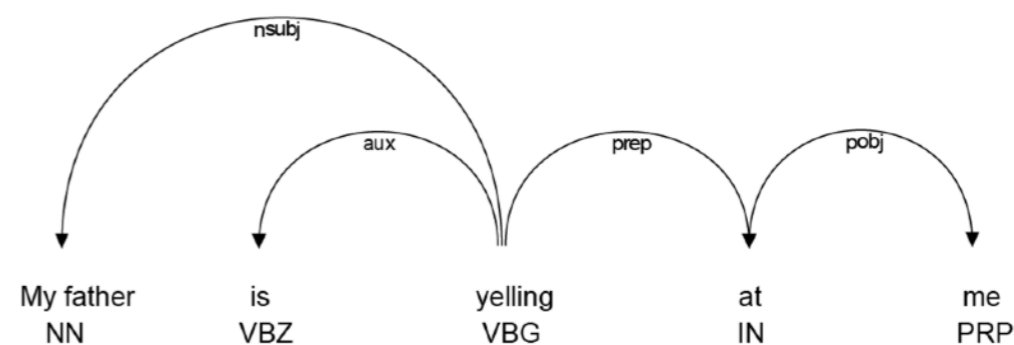
Algorithm

1. We extract a series of **mentions** potentially referring to real world entities
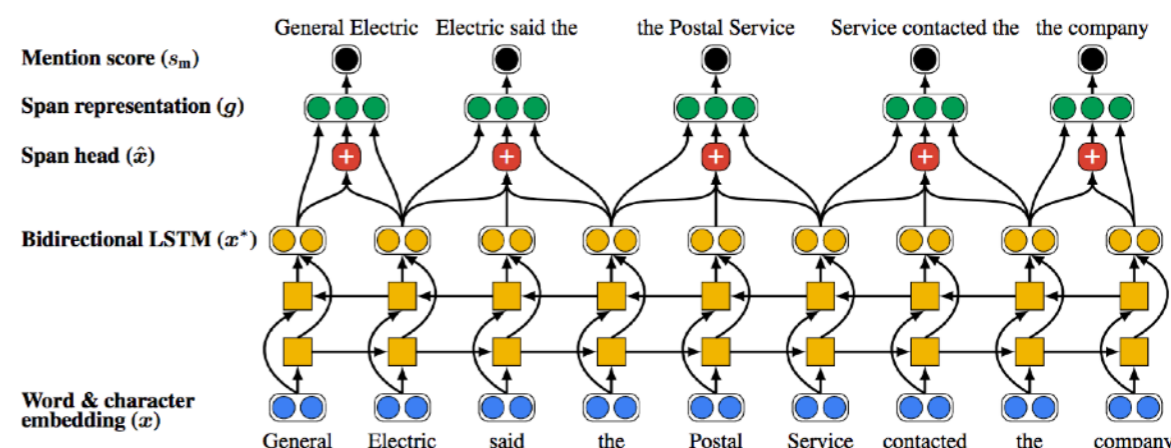
- **Rule-based** (our approach):

Parse the input and apply a set of rules to extract segments of the sentence => 90% recall

- **Neural-network-based** (Lee et al. EMNLP 2017):

Train a neural net to score potential segments of the sentence.

Algorithm

2. For each pair of mentions, we compute a set of about twenty **features**:

- **word embeddings** in/around each mention,

- **distance** between mentions,

- **boolean features** related to the speakers in dialog (same speakers, exact string match)

3. We find the most likely antecedent for each mention by comparing each pair => **pairwise ranking**
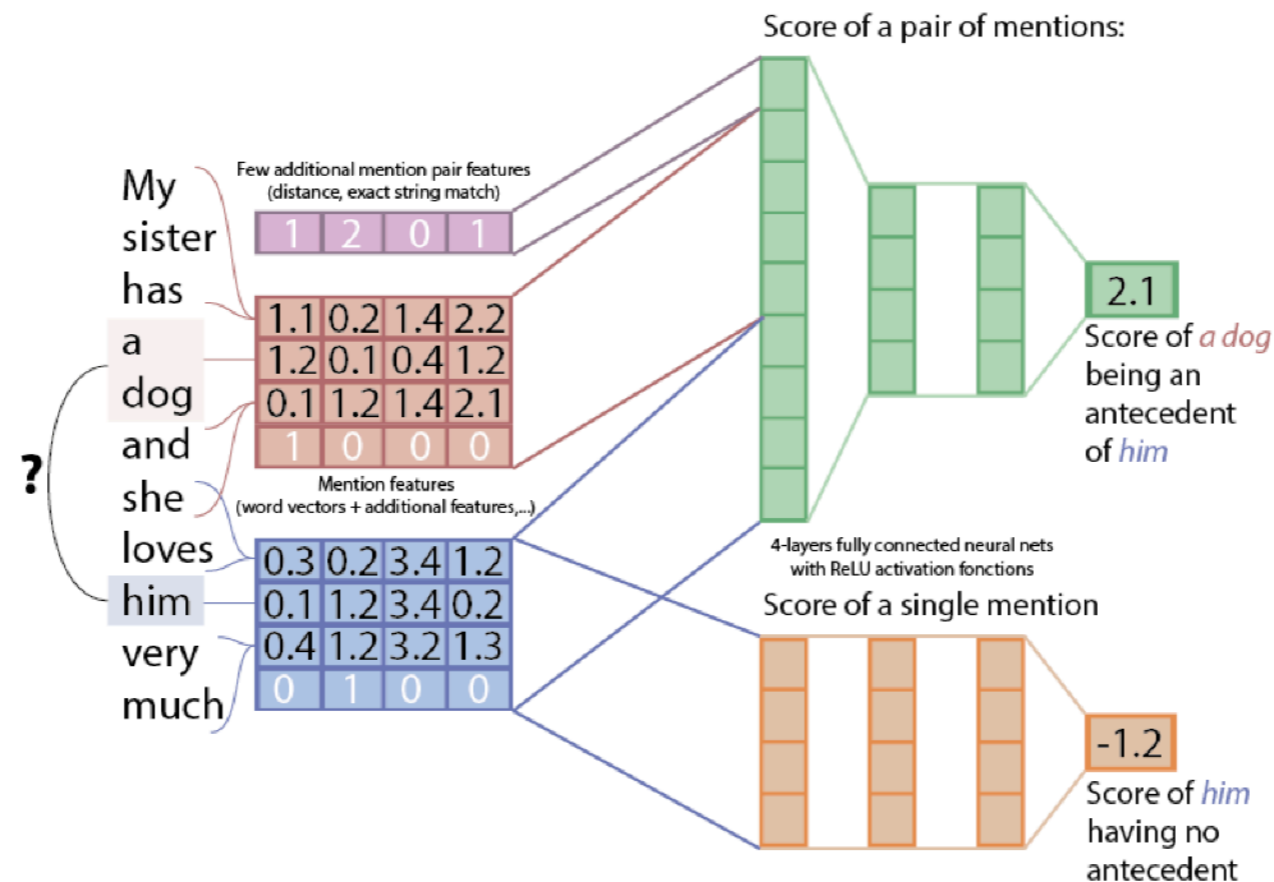
- Network is pre-trained with maximum likelihood and trained on a non-probabilistic slack-rescaled max-margin objective

Trained features
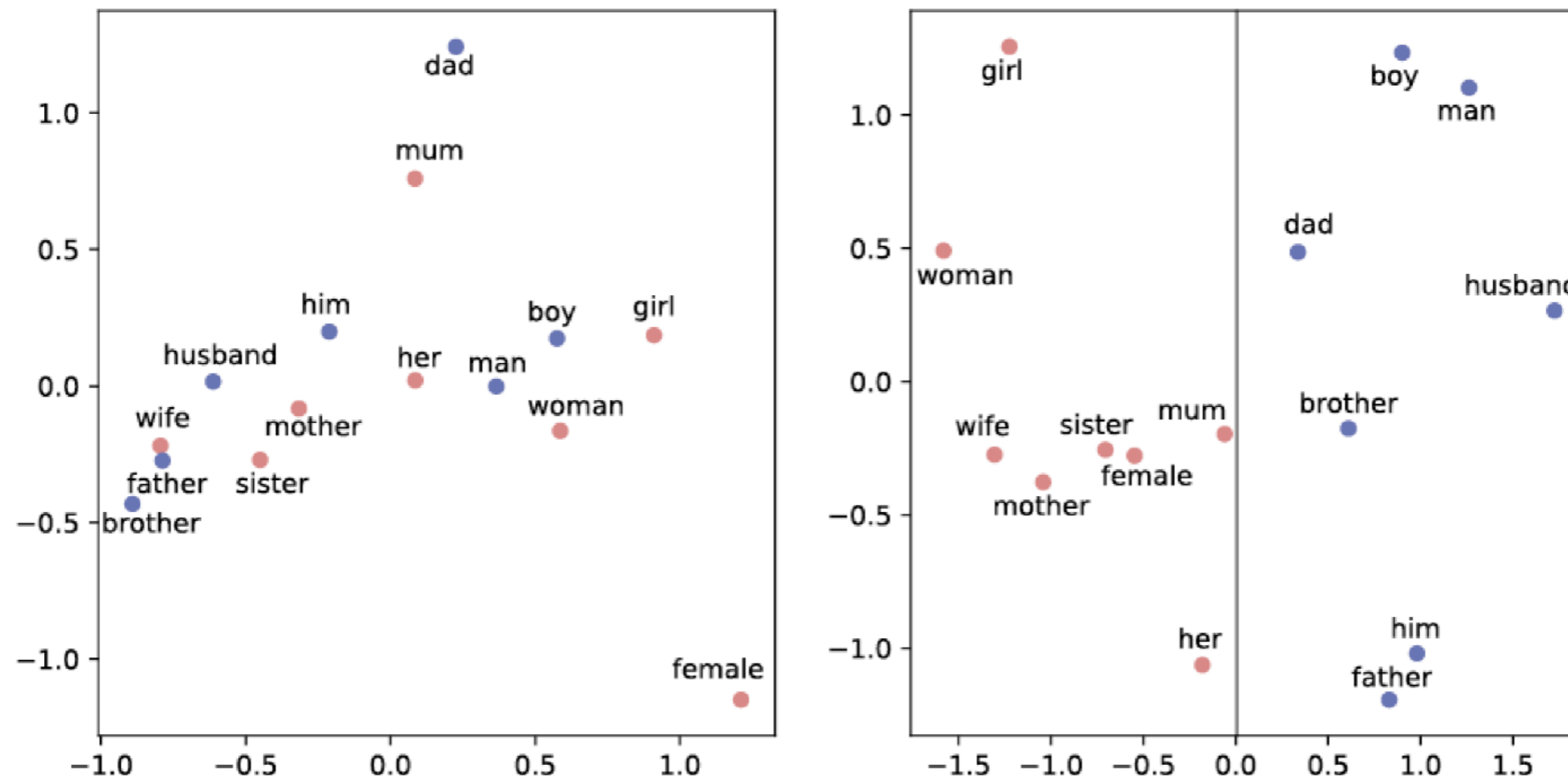
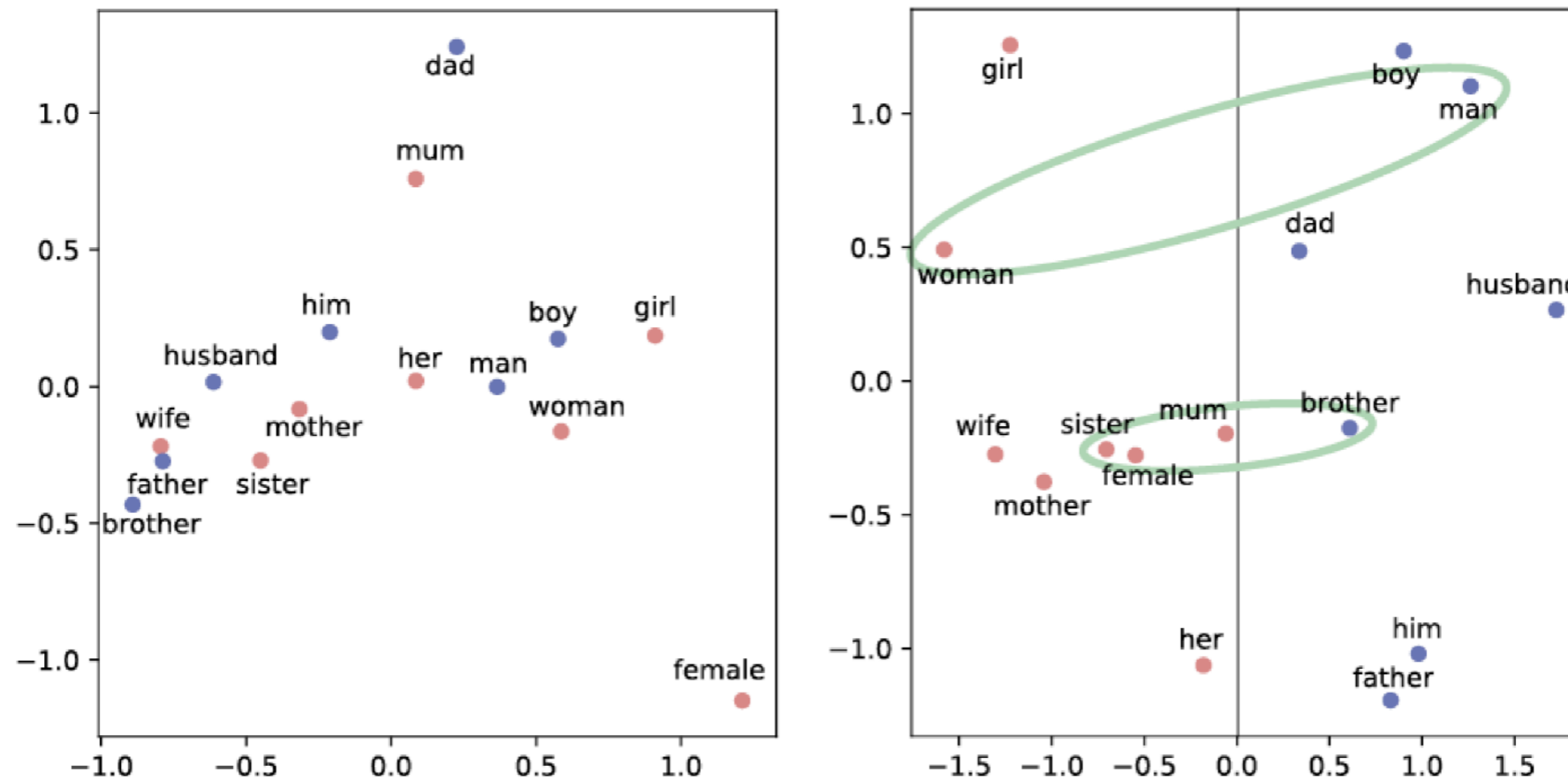
- Left: **Initial** word embeddings (PCA of pre-trained word2vec)

- Right: **trained** word embeddings (PCA)

Trained features

- Left: **Initial** word embeddings (PCA of pre-trained word2vec)

- Right: **trained** word embeddings (PCA)

- Trained on OntoNotes Corpus – formal language

Open-sourced
for the conversational agents community

- Interesting versus alternative coreference solutions (Stanford's CoreNLP)

  - Modular Python module VS monolithic Java bloc => easier to integrate in high-throuput distributed systems

  - Can makes use of speakers informations in a dialog => better performances in dialog systems

  - Easily adapt to evolving vocabulary: compute embeddings for unknown words on the fly from definitions => better performance in challenging language field (teenage language, slang, …)

  - Based on spaCy ultra-fast cython/python parser => Numpy/pyTorch style, pythonic approach

- Get it on https://github.com/huggingface and try it at: https://huggingface.co/coref/