**Representation Learning**

In machine learning, vector representations are now widely appreciated for their ability to impart background information about items into a learning process. For example, the machine learning algorithm may never have seen the string *Havel* in its training data, but if a vector representation for *Havel* reveals that it is similar to other European rivers such as *Elbe* and *Danube*, the machine learning algorithm may be able to treat it appropriately.

**Semantic Tag Vectors**

- Most word vector representations contain numbers that are not human-interpretable. We consider vectors that explicitly store information about a word’s semantic meaning properties.
- Semantic tagging (Bjerva et al. 2016) is a recently proposed scheme for labeling the properties of words. E.g.:

  **The dog attacked the little boy.**

  - **The**
    - DEFINITE
  - **dog**
    - DEFINITE
  - **attacked**
    - EPS: past simple: ate, went
  - **the**
    - DEFINITE
  - **little**
    - INTERACTIVE: open, vegetarian, quickly
  - **boy**
    - DEFINITE

- We create a vector for a word, in which each dimension captures how often we saw it labeled with a specific tag in a human-labeled text collection.

- **Our Goal**: Given these vectors, which we only have for a small set of words for which we have human-provided tags, automatically create similar semantic tag vector representations for new words.

---

**Methodology**

**Generating Semantic Tag Vectors from Labeled Text Collection (PMB Dataset)**

- One-hot vector for each appearance of token \( \omega \) storing which tag it is labelled with.
- \( \sum \) Each position corresponds to a unique semantic tag.

**Find k Nearest Neighbors**

- Input Word
- Find most similar words using Sketch Engine word Embeddings.
- But importantly, only consider entries with same part-of-speech (e.g., verb vs. noun)
- Example: Notice-v → see-v, spot-v

**Predict Semantic Tag Vector**

- Combine k vectors by taking average weighted by cosine similarity scores between \( \omega \) and \( \omega_i \).