Low-Resource NLP

David R. Mortensen

Algorithms for Natural Language Processing

Learning Objectives

- Know what a low-resource language or domain is
- Know three main approaches to low-resource NLP:
  - Traditional/rule based
  - Unsupervised learning
  - Transfer learning
- Know three examples of transfer learning

Low-Resource Natural Language Processing

- Carrying out NLP tasks for…
  - languages…
  - domains…
- without…
  - parallel corpora
  - extensive monolingual corpora
  - other annotated data
  - existing NLP tools

Most NLP Problems are Low-Resource NLP Problems

- **Most languages are low-resource**
  - Approximately 7,000 languages
  - Adequate NLP resources for about 10 languages
  - Most people in the world speech a language not included in that 10
- **Most domains are low-resource**
  - Biomedical text
  - Legal text
  - Literary text
  - Twitter
- **Solving any of these problems requires doing low-resource NLP**

Approaches

Traditional

- Get more data
- Build language-specific tools with linguistic knowledge

Unsupervised learning

- Use machine learning techniques that do not require labeled training data

Transfer

- Exploit training data from higher-resource settings to provide supervision for low-resource scenarios

Traditional Approaches

Obtaining More Data

- The naivest approach to low-resource scenarios is to convert them to high-resource scenarios
  - Obtain more unannotated data
  - Annotate it
- This has a number of obvious shortcomings
  - Raw data is often difficult to obtain.
    - Domains where only a limited amount of text exists, like law or medicine
    - Languages that do not have a significant internet presence
  - Annotation of data is expensive
    - Turkers are cheap, but unskilled and still cost money
    - Experts are expensive and slow

Rule-Based NLP

- One approach to low-resource NLP is to use models that are based on linguistic descriptions rather than being data-driven
- Given a reference grammar of sufficient quality and a lexicon, a computational linguist can build rule-based models for many things:
  - Morphological analysis
  - Parsing
  - Named entity recognition
  - Relation extraction
- However, this is also problematic
  - Not enough grammars
  - Not enough computational linguists

Linguistically Inspired ≠ Rule Based

- However, using linguistic knowledge does not mean constructing an entirely rule-based system
- One successful approach:
  - Combine linguistic knowledge and machine learning
  - Not easy with deep learning, but possible
  - For examples, stay tuned

Unsupervised Approaches

Not All Machine Learning is Supervised

- Suppose you have a large body of unlabeled data, but little or no labeled data
- You can extract a lot of patterns from it
- For example, word embeddings and models like BERT are unsupervised
- Human language learning is also largely unsupervised (although we do get some supervision for other senses) so we know it is possible to learn language without labeled data

Brown Clusters

- **Hierarchical agglomerative** clustering of words based on the contexts in which they occur
- Purely unsupervised
- Semantically related words end up in the same part of the tree
    - City names cluster together
    - Country names cluster together
    - Colors cluster together
- **Example from SLP**: suppose you want to know the probability of “to Shanghai” but the bigram “to Shanghai” never occurs in the data. You can estimate the probability by looking “to X” where X is other city names in the same cluster with Shanghai.

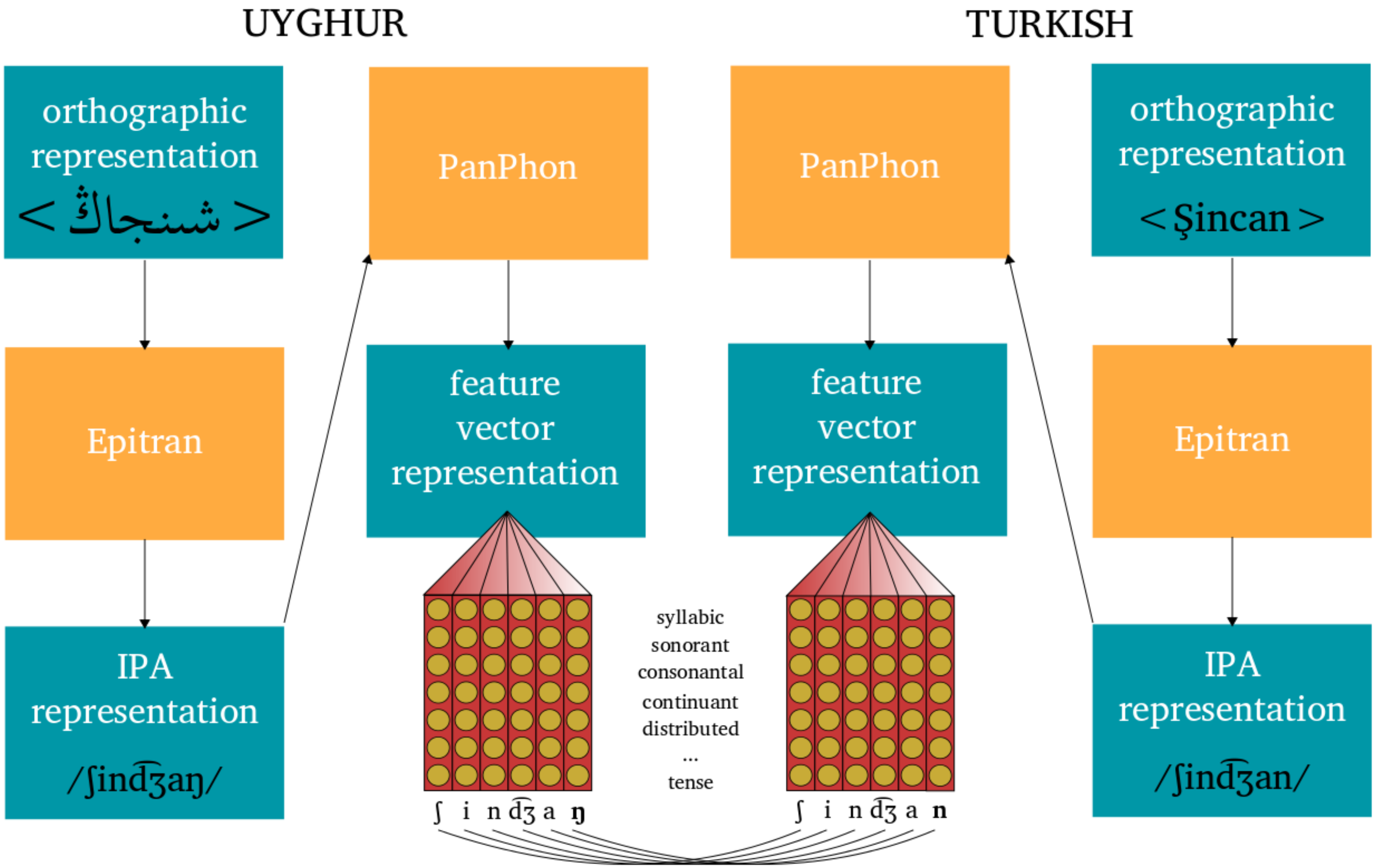
Transfer Learning

Learn One Place, Apply Elsewhere

- As humans, we have little problem generalizing knowledge gained in one domain to other domains
  - When we are reading legal documents, we use knowledge that we gained reading everyday English
  - When we learn Japanese, we may use knowledge that we gained speaking Korean
- This is the basic idea behind **transfer learning**
- It involves techniques to “transfer” knowledge gained in one domain to another

One Example: Uyghur NER

- Uyghur is a low-resource language spoken in the northwest of China.
- It is related to other, higher-resource, languages like Uzbek, Kazakh, Turkmen, and Turkish
- Turkish, Uzbek, and Uyghur are each written with a different script
- We built a Uyghur NER model as follows:
  - Convert all of the data to IPA (the International Phonetic Alphabet)
  - Convert IPA to articulatory features (phonetic features that define how each sound is produced)
  - Train a model on Turkish and Uzbek
  - Tune the model on Uyghur, and test on Uyghur



Bharadwaj, A., Mortensen, D. R., Dyer, C., & Carbonell, J. G. (2016, November). Phonologically aware neural model for named entity recognition in low resource transfer settings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1462-1472).

Another Example: Cross-Lingual Dependency Parsing

- Interested parties have now produced a large collection of dependency treebanks called the Universal Dependency (or UD) Treebanks
- Dependency trees have a lot in common between languages
  - This commonalities are often latent structures
  - Related languages tend to have more shared structural properties than randomly selected languages
- It is possible to train cross-lingual or polyglot dependency parsers and to use them on languages for which there is no treebank
- Lots of techniques for this