I320D – Topics in Human Centered Data Science
Text Mining and NLP Essentials

Week 5: Grammar and languages, Representing syntax, Part of Speech Tagging, Shallow and Deep Parsing, Constituency and Dependency parsing

Dr. Abhijit Mishra

Week 4: Recap

- **Lecture:**
  - Representing Words, Sentences and Documents,
  - Bag-of-words, N-grams, TF-IDF , Introduction to Word Vectors (GloVE)
  - Document similarity and distance metrics
- **Lab:**
  - Document Representation and Similarity Measurement
  - Building Semantic Search systems

Ongoing and upcoming assignment

- **Upcoming:** Semantic Search of Tweets (to be posted tonight)
  - Deadline: Thursday, 02/22

So far in I320D – Text Mining and NLP

- W1. Language and Ambiguity
- W2. Basics of Text Data and Linguistic Concepts
- W3. Text Preprocessing Techniques
- W4. Lexical Analysis
- W5. Syntax Analysis
- W6. Information Extraction

W7. Machine Learning Methods for NLP
W8. Unsupervised ML and Topic Modeling Basics
W10-W11. Deep learning for NLP
W12. NLP Applications
W13. Small and Large Language Models and Prompt Engineering Basics
W14. Knowledge Networks
W15. Evaluation Metrics

Recap: NLP Tasks and the Need for Representing Documents

FEATURIZE
or
Extract a computer understandable, mathematically sound and linguistically viable form

- **Classification**
- **Search**

Document Classification

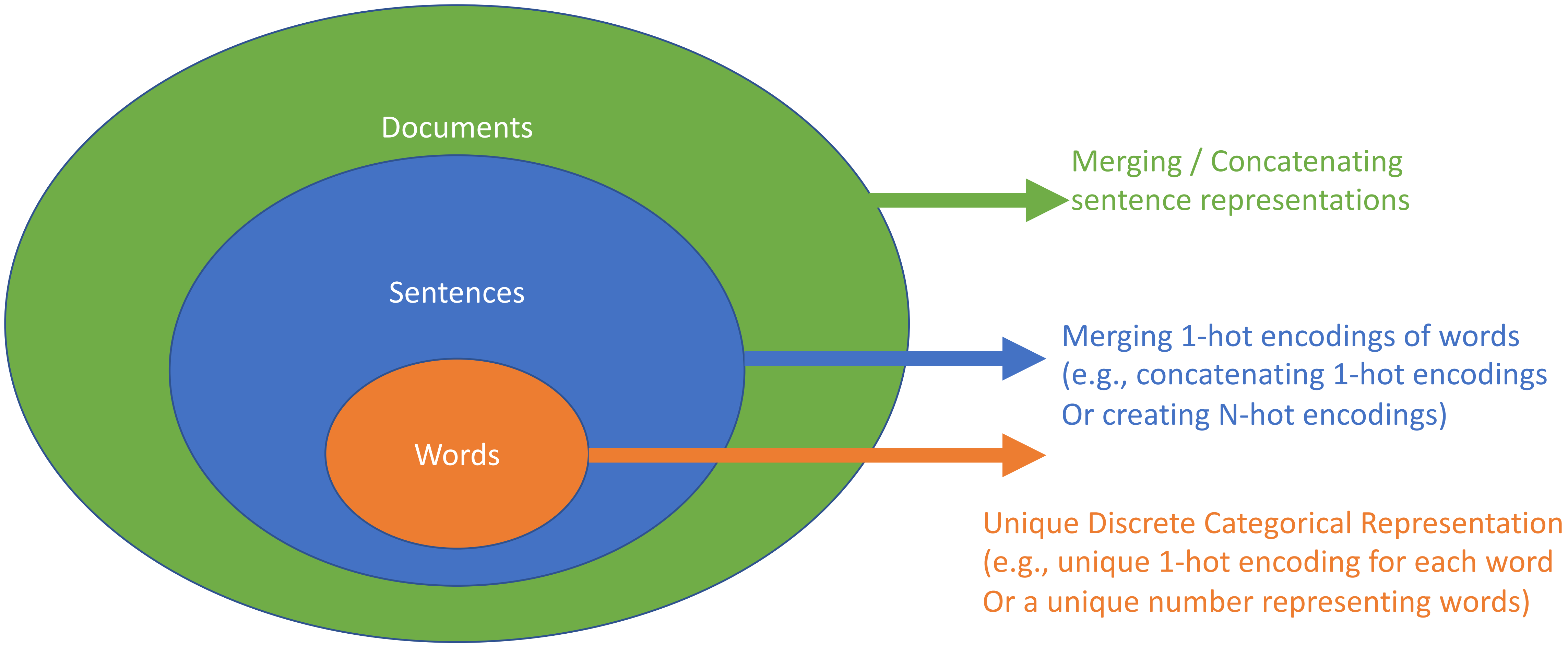
Token Classification (sequence labeling)

Text Generation (sequence to sequence)

Document Similarity Computation and Ranking

* Text generation is a special case of text classification

Recap: Word / Sentence / Document Representations



How to Represent Words in Documents

- **Sparse** vectors:
  - 1-hot vectors or Bag-of-words (presence/absence or count)
  - Term-frequency – Inverse Document Frequency (TF-IDF)
- **Dense** Word vectors learned using unlabeled corpora
  - Matrix Factorization based (e.g., Latent Semantic Analysis)
  - Neural Network based (Word2Vec, Glove)

Recap: Text Pre-processing for Data Sparsity Reduction

- Cleaning and normalization (remove noise)
- Tokenize Text
- Perform Morphological Analysis
- Remove Stopwords
- Extract Representations (BoW, GloVE)

Questions?

Quiz: Question 1

- What is the main difference between stemming and lemmatization?
  - a) Stemming only removes suffixes, while lemmatization reduces words to their base form.
  - b) Stemming converts words to lowercase, while lemmatization maintains the original case.
  - c) Stemming and lemmatization are the same.
  - d) Stemming is faster than lemmatization.

a) Stemming only removes suffixes, while lemmatization reduces words to their base form.

Quiz: Question 2

In one-hot vectorization, how is the length of the vector determined?

a) It is equal to the number of unique words in the corpus.
b) It is equal to the length of the longest document in the corpus.

c) It is equal to the number of documents in the corpus.

d) It is equal to the number of sentences in the corpus.

Answer: a) It is equal to the number of unique words in the corpus.

Quiz: Question 3

1. Which vectorization technique represents each document as a vector of word frequencies?
    a) One-hot vectorization
    b) b) Count vectorization
    c) c) Word embeddings
    d) d) TF-IDF vectorization

Answer: b) Count vectorization

Quiz: Question 4

1. Which of the following methods captures semantic meaning of words and their relationships in a continuous vector space?
   a) One-hot encoding
   b) Count vectorization
   c) c) Word embeddings
   d) d) TF-IDF vectorization

   Answer: c) Word embeddings

Quiz: Question 5

1. Which of the following techniques represents each document as a vector in a high-dimensional space, where each dimension corresponds to a unique word?
    1. a) One-hot vectorization
    2. b) Count vectorization
    3. c) Word embeddings
    4. d) TF-IDF vectorization

Answer: a, b, d

Week 5 : Syntax Analysis

- **Lecture:**
  - Grammar and Languages
  - Representing Syntax
  - Part of Speech Tagging
  - Shallow and Deep Parsing
  - Constituency and Dependency parsing
- **Lab:**
  - Leveraging Syntax Analysis + RegEx for pattern extraction

Syntax Analysis

- "Syntax analysis, also known as ***parsing***, is a crucial step in NLP that involves the "analysis of the grammatical structure of a sentence or text in order to understand its syntactic relationships and hierarchies."

Two Kinds

- Shallow parsing
- Deep parsing

Shallow Parsing

- Shallow parsing, also known as **tagging, chunking** or **partial parsing**
- Focuses on identifying and grouping words or phrases in a sentence into larger syntactic units,
    - often without establishing the full syntactic relationships between them.

Shallow Parsing Tasks

- Part of Speech Tagging
- Noun Phrases / Verb Phrases Chunking
- Named Entity Identification
- Multiword Detection

Part of Speech Tagging

- A kind of shallow parsing is that involves assigning a specific part of speech to each word in a sentence or text.
- **Objective:**
  - **Input:** A sequence of tokens $w_1, w_2, \ldots, w_N$ constituting a sentence S
  - **Output:** For each word w_i , assign a part-of-speech tag t_i from a predefined set of tags (e.g., noun, verb, adjective, adverb, etc.)**, also known as Tagset.**

PoS Examples

```
# Example 1
sentence_1 = ["<START>", "The", "cat", "is", "sleeping.", "<END>"]
tags_1 = ["<START>", "DT", "NN", "VBZ", "VBG", "<END>"]
# Example 2
sentence_2 = ["<START>", "She", "sells", "seashells", "by", "the", "seashore.", "<END>"]
tags_2 = ["<START>", "PRP", "VBZ", "NNS", "IN", "DT", "NN", "<END>"]
# Example 3
sentence_3 = ["<START>", "The", "quick", "brown", "fox", "jumps", "over", "the", "lazy",
"dog.", "<END>"]
tags_3 = ["<START>", "DT", "JJ", "JJ", "NN", "VBZ", "IN", "DT", "JJ", "NN", "<END>"]
```

Why PoS Tagging?

- A crucial component in higher layers of NLP processing
  - Constituency and Dependency Parsing
  - Named Entity Identification
- Useful in pattern extraction
  - **Example:** Healthcare
    - Identifying medical conditions mentioned in patient records, such as "*diabetes mellitus type 2" (POS pattern: NN NN NN CD)*
  - **Example:** Finance
    - Recognizing financial terms in news articles, such as "stock market", "interest rate", or "bond yield", by identifying noun phrases (NP) with specific patterns like "NN + NN" or "JJ + noun".

Tagset

- Predefined set of tags or labels used to annotate words in a corpus or dataset with their corresponding parts of speech or grammatical categories
- Tags are often abbreviated in 2-3 capital letters
- Example :
  - Penn tagset for English
  - https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Tagset (...)

1. **Noun (NN):** Nouns are words that represent people, places, things, or concepts.
   - Examples: "dog," "cat," "book," "house," "love"
2. **Verb (VB):** Verbs denote actions, states, or occurrences.
   - Examples: "run," "eat," "read," "is," "am"
3. **Adjective (JJ):** Adjectives describe or modify nouns by providing more information about them.
   - Examples: "happy," "red," "tall," "beautiful"
4. **Adverb (RB):** Adverbs modify verbs, adjectives, or other adverbs by providing information about how, when, or where an action takes place.
   - Examples: "quickly," "very," "now," "here"
5. **Pronoun (PRP):** Pronouns replace nouns in sentences to avoid repetition.
   - Examples: "he," "she," "it," "they," "we"
6. **Preposition (IN):** Prepositions show the relationship between nouns (or pronouns) and other words in a sentence.
   - Examples: "in," "on," "at," "with," "under"
7. **Conjunction (CC):** Conjunctions connect words, phrases, or clauses.
   - Examples: "and," "but," "or," "because," "although"
8. **Interjection (UH):** Interjections express strong emotions or exclamations.
   - Examples: "wow," "ouch," "oh," "hurray"
9. **Determiner (DT):** Determiners are used to specify or clarify a noun.
   - Examples: "the," "a," "an," "this," "some"

Different Tagsets for Different Languages

Tagsets in English may differ from those in other languages due to variations in grammatical structures, linguistic features, and the specific needs of natural language processing (NLP) tasks.

So, tagsets are often language specific to capture language specific grammatical represntations

Different Tagsets for Different Languages

- **Grammatical Categories:** Different languages may have different sets of grammatical categories or parts of speech.
- Languages such as Spanish or French may have additional categories like articles (definite and indefinite), clitics, or pronominal adverbs.
  - Me gusta el café." (I like coffee.)
  - Here "Me" would need a specific tag ”PRN_CLIT".
- Indic languages often exhibit compounding through specific rules and case-markers attached to words

Indic Tagset

Table 1. LDC-IL tagset

Category	Types	Attributes
Noun	Common (NC) Proper (NP) Verbal (NV) Spatio-Temporal (NST)	Gender, Number, Case, Distributive, Honorificity, Emphatic dimension
Pronoun	Pronominal (PR) Reflexive (RF) Reciprocal (RC) Relative (RL) Wh (WH)	Gender, Number, Person, Case, Case marker, Distributive, Emphatic, Dimension, Honorificity
Demonstrative (D)	Absolutive (DAB) Relative Demonstrative (DRL) Wh-Demonstrative (DWH)	Number, Case, Dimension, Distributive, Emphatic (not in case of wh)
NominalMod	Adjectives (JJ)	Gender, Number, Case,

ifier (J)	Quantifiers (JQ) Intensifier (JINT)	Numeral, Distributive
Verb(V)	Main verb(VM) Auxiliary verb(VA)	Gender, Number, Person, Tense, Aspect, Mood, Finiteness ,Honorificity
Adverb(A)	Manner(AMN)	Case, Distributive
Post-Position(PP)		Gender, Number, Case marker
Numeral (NUM)	Real (NUMR) Serial (NUMS) Calendric (NUMC) Ordinal (NUMO)	
Residual(RD)	Foreign Word (RDF) Symbol (RDS)	
Unknown		
Punctuation(PU)		

https://www.digitalxplore.org/up_proc/pdf/55-139590032413-17.pdf

Mathematical Formulation of Shallow Parsing

- Let’s consider PoS Tagging example

Sequence Labeling Foundations:

$W = [w_1, w_2, \ldots w_N]$ $\longrightarrow$ $T = [t_1, t_2, \ldots t_N]$

Sequence W of N tokens is transformed into sequence T of N tags

$$\mathbf{T}^* = \mathbf{argmax}_{\mathbf{T}}\, \boldsymbol{p}(\mathbf{T}|\mathbf{W})$$
$$= \mathbf{argmax}_{\mathbf{T}}\, p(t_1, t_2, \ldots t_N | w_1, w_2, \ldots w_N)$$

Digression: Probability Primer

- Probability of an event is a number indicating how likely that event will occur
  - E.g., For any unbiased coin, $p(x = \text{“}head\text{”}) = p(x = \text{“}tail\text{”}) = 0.5$
- Highly related with frequency of occurrence

$$p(E) = \frac{n(E)}{n(S)}$$

Where

- $p(E)$ is the probability of event E.
- n(E) is the number of favorable outcomes (outcomes where event *E* occurs).
- n(S) is the total number of possible outcomes in the sample space.

Digression: Probability Primer (1)

- Conditional probability:
  - Where probability of an event is ”conditioned” on a set of observations
  - For any unbiased coin, $p(x = \text{“}head\text{”}) = p(x = \text{“}tail\text{”}) = \; 0.5$

  - But, if we observe that the last 5 tosses have given HHTTH, we can say

$$p(x = \text{“}head\text{”} \mid ovservations = HHTHH) = \frac{3}{5}$$

  - This is sometimes referred to as **likelihood** of an event
  - In predictive analysis, we often predict based on likelihood

Digression: Probability Primer (3)

- **Maximum Likelihood Estimation (MLE)** is a statistical technique that seeks to find the parameter values in a statistical model that maximize the likelihood of the observed data.
- Commonly used for parameter estimation in various fields and involves maximizing the likelihood function to obtain parameter estimates.

Digression: Max and ArgMax

- Argmax : a term used to find the input value that gives the highest output value in a given set of options
- Say $x = [5, 12, 8, 20, 15]$

  max (x) = 20

  argmax (x) = 4 (index of 20), starting from index 1

POS Tagging

- POS Tagging: attaches to each word in a sentence a part of speech tag from a given set of tags called the Tag-Set
- *Standard Tag-set : Penn Treebank (for English).*
- *Example:*

  “_“ The_DT mechanisms_NNS that_WDT
  make_VBP traditional_JJ hardware_NN
  are_VBP really_RB being_VBG
  obsoleted_VBN by_IN microprocessor-
  based_JJ machines_NNS ,_, ”_” said_VBD
  Mr._NNP Benton_NNP ._.

POS Tagging as an MLE Estimate

- Consider example

  "People jump high"

- **Hypothetical Tagset: ["N", "V", "A"]**
- **Possible tags for each token**
  - **People :** Noun **(N),** Verb **(V)**
  - **Jump:** Noun **(N),** Verb **(V)**
  - **High:** Adjective **(A),** Noun **(N) (say we ignore adverb)**

POS Tagging as an MLE Estimate (1)

- Possible output sequences
    - *People_N jump_N high_N*
    - *People_V jump_N high_N*
    - *People_N jump_V high_N*
    - *People_V jump_V high_N*
    - *People_N jump_N high_A*
    - *People_V jump_N high_A*
    - *People_N jump_V high_A*
    - *People_V jump_V high_A*
    - *People_N jump_N high_N*
    - *People_V jump_N high_N*
    - *People_N jump_V high_N*
    - *People_V jump_V high_N*
    - …
    - …

POS Tagging as an MLE Estimate

- Say we have a scoring mechanism (that's our model)

$$p(\mathrm{t}_1 = \text{“N”}, \mathrm{t}_2 = \text{“N”}, \mathrm{t}_3 = \text{“}N\text{”} | \mathrm{w}_1 = \text{“}People\text{”}, \mathrm{w}_2 = \text{“jump”}, w_3 = \text{”}high\text{”}) = 0.01$$
$$p(\mathrm{t}_1 = \text{“N”}, \mathrm{t}_2 = \text{“V”}, \mathrm{t}_3 = \text{“}A\text{”} | \mathrm{w}_1 = \text{“}People\text{”}, \mathrm{w}_2 = \text{“jump”}, w_3 = \text{”}high\text{”}) = 0.8$$
$$p(\mathrm{t}_1 = \text{“N”}, \mathrm{t}_2 = \text{“V”}, \mathrm{t}_3 = \text{“}R\text{”} | \mathrm{w}_1 = \text{“}People\text{”}, \mathrm{w}_2 = \text{“jump”}, w_3 = \text{”}high\text{”}) = 0.001$$
$$p(\mathrm{t}_1 = \text{“N”}, \mathrm{t}_2 = \text{“N”}, \mathrm{t}_3 = \text{“}A\text{”} | \mathrm{w}_1 = \text{“}People\text{”}, \mathrm{w}_2 = \text{“jump”}, w_3 = \text{”}high\text{”}) = 0.003$$
$$p(\mathrm{t}_1 = \text{“V”}, \mathrm{t}_2 = \text{“N”}, \mathrm{t}_3 = \text{“}A\text{”} | \mathrm{w}_1 = \text{“}People\text{”}, \mathrm{w}_2 = \text{“jump”}, w_3 = \text{”}high\text{”}) = 0.02$$

...

- The winner tag sequence will be $\mathrm{T}^* = \mathrm{argmax}_{\mathrm{T}}\, p(\mathrm{t}_1, \mathrm{t}_2, \mathrm{t}_3 | \mathrm{w}_1, \mathrm{w}_2, \mathrm{w}_3)$ for all possible combinations of Ts and Ws
- Here, $\mathrm{T}^* = [\mathrm{N}, \mathrm{V}, \mathrm{A}]$
- In other words, ***“which sequence of T maximizes the likelihood”***

So, it's all about defining the scoring mechanism (that's our model)

$$\mathbf{T}^* = \mathbf{argmax}_{\mathrm{T}}\, \boldsymbol{p}(\mathbf{T}|\mathbf{W})$$
$$= \mathbf{argmax}_{\mathrm{T}}\, p(\mathrm{t}_1, \mathrm{t}_2, \ldots \mathrm{t}_{\mathrm{N}}|\mathrm{w}_1, \mathrm{w}_2, \ldots \mathrm{w}_{\mathrm{N}})$$

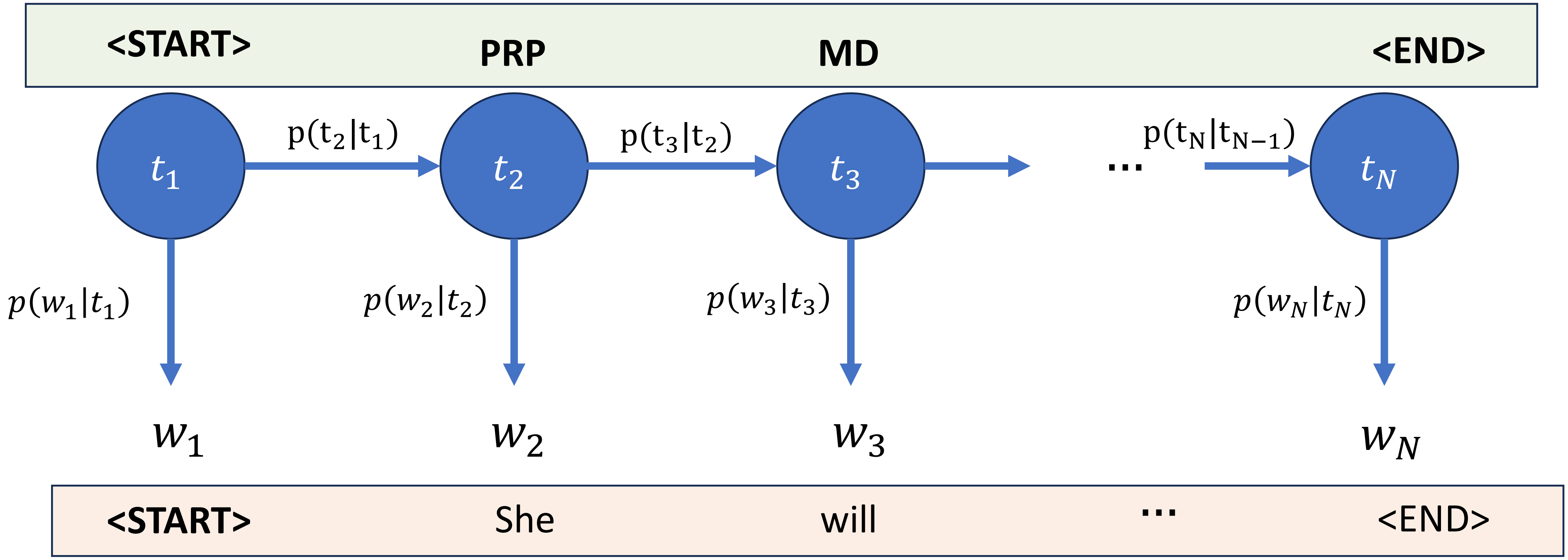
That's our model …

Source of Knowledge for scoring – Training Data

- A tagged dataset – or **POS Annotated Corpus**
- **Example POS Corpus:**
  - Penn Treebank Tagset (PTB Tagset)
  - Tweet tagset(CMU)
  - Brown Corpus Tagset
  - Google Universal POS Tagset
  - Indic POS Corpus

$p(\mathbf{T}|\mathbf{W})$ - Possible Solution 1 - HMM

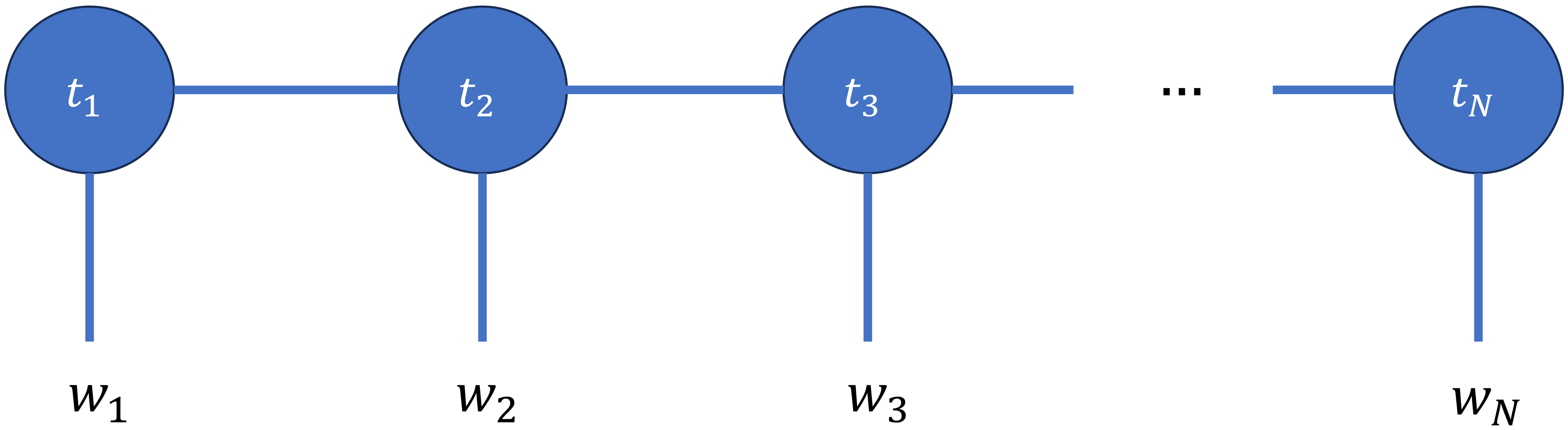
Example: She will bear the burden



$$\mathbf{argmax_T}\ \boldsymbol{p}(\mathbf{T}|\mathbf{W}) = \mathbf{argmax_T} \prod_{i=1}^{N} \mathrm{p}\,(w_i|t_i)\ \mathrm{p}(\mathrm{t_i}|\mathrm{t_{i-1}})$$

Also Known as "Hidden Markov Model" formulation

$p(\mathrm{T}|\mathrm{W})$ - Possible Solution 2 - CRF



$$\mathbf{argmax_T}\ \boldsymbol{p}(\mathbf{T}|\mathbf{W}) = \mathbf{argmax_T} \prod_{i=1}^{N} \mathrm{p}\left(t_i | t_{i-1}, w_1, w_2, \dots, w_{t-1}\right)$$

Where,

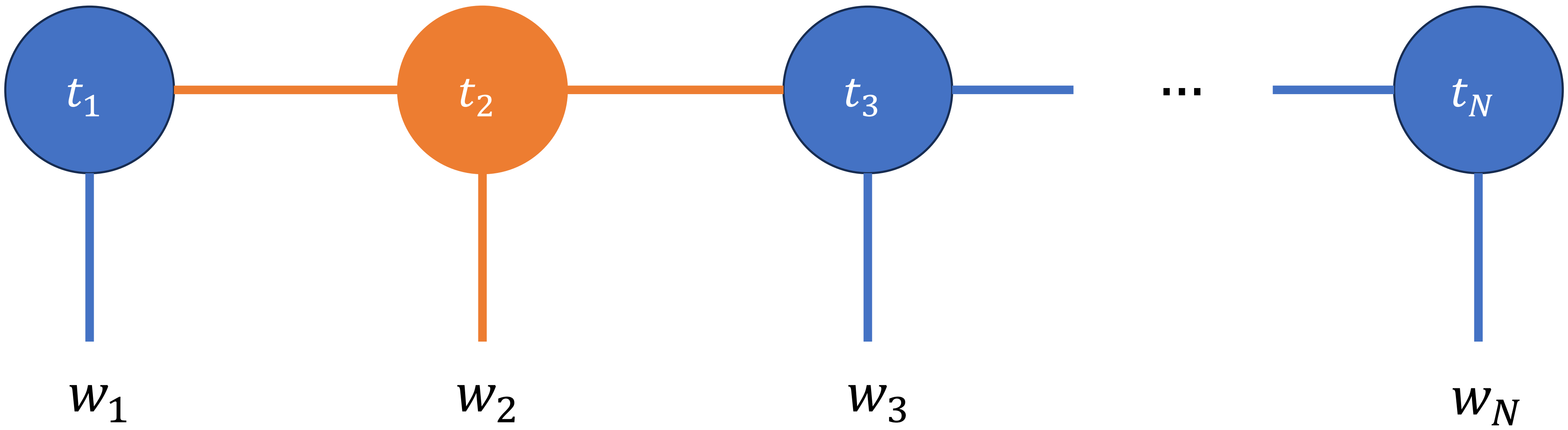
$$\mathrm{p}(t_i | t_{i-1}, w_1, w_2, \dots, w_{t-1}) = \frac{1}{Z} \exp\left(\theta_1 f_1(\mathrm{t_i}, \mathrm{t_i} - 1) + \theta_2 f_2(t_i, w_1, w_2, \dots w_N)\right)$$

f1, f2 are features . Thetas are weights and Z is a normalization constant

Also Known as "Conditional random field" formulation

No Arrows / Directionality:
Each tag is dependent on all of it's neighborhood

$p(\mathrm{T}|\mathrm{W})$ - Possible Solution 2 - CRF



$$\mathbf{argmax_T}\ \boldsymbol{p}(\mathbf{T}|\mathbf{W}) = \mathbf{argmax_T} \prod_{i=1}^{N} \mathrm{p}\left(t_i | t_{i-1}, w_1, w_2, \dots, w_{t-1}\right)$$

Where,

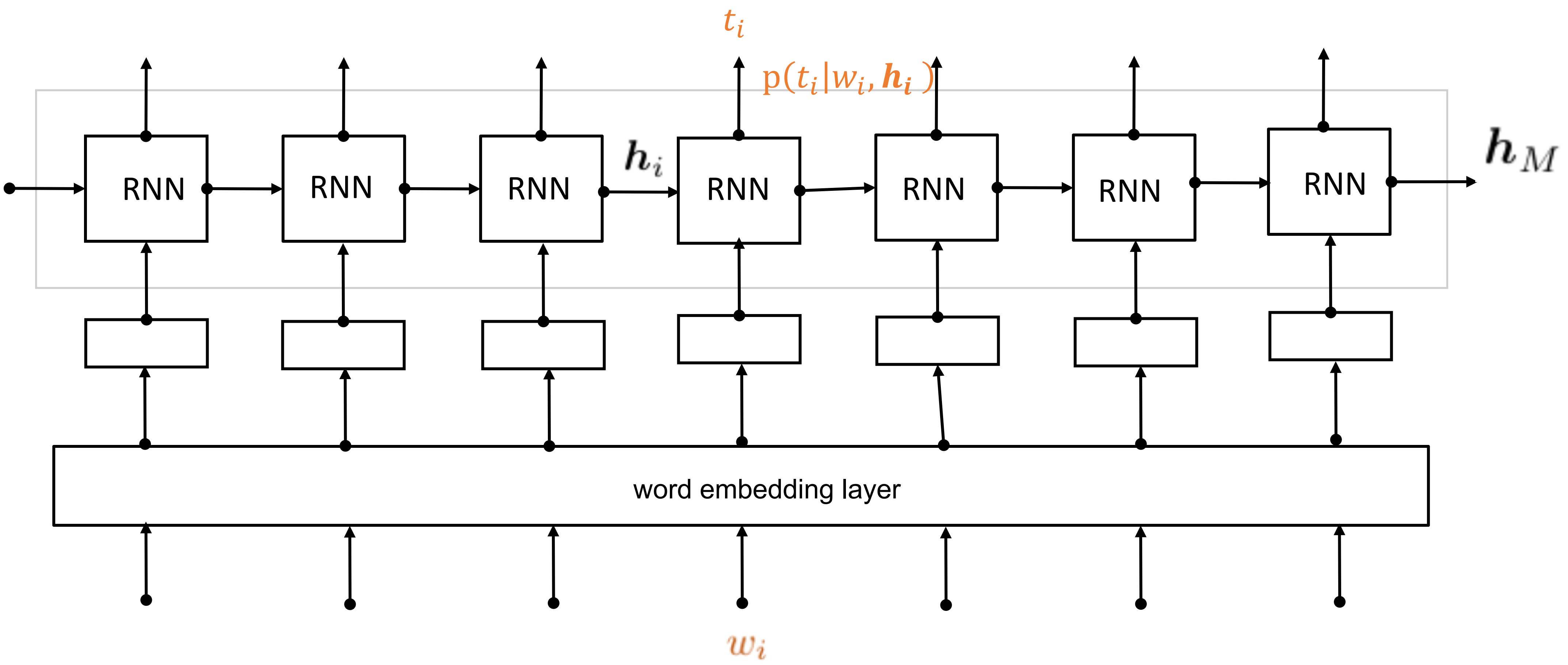
$$\mathrm{p}(t_i | t_{i-1}, w_1, w_2, \dots, w_{t-1}) = \frac{1}{Z} \exp\left(\theta_1 f_1(\mathrm{t_i}, \mathrm{t_i} - 1) + \theta_2 f_2(t_i, w_1, w_2, \dots w_N)\right)$$

f1, f2 are features . Thetas are weights and Z is a normalization constant

Also Known as “Conditional random field” formulation

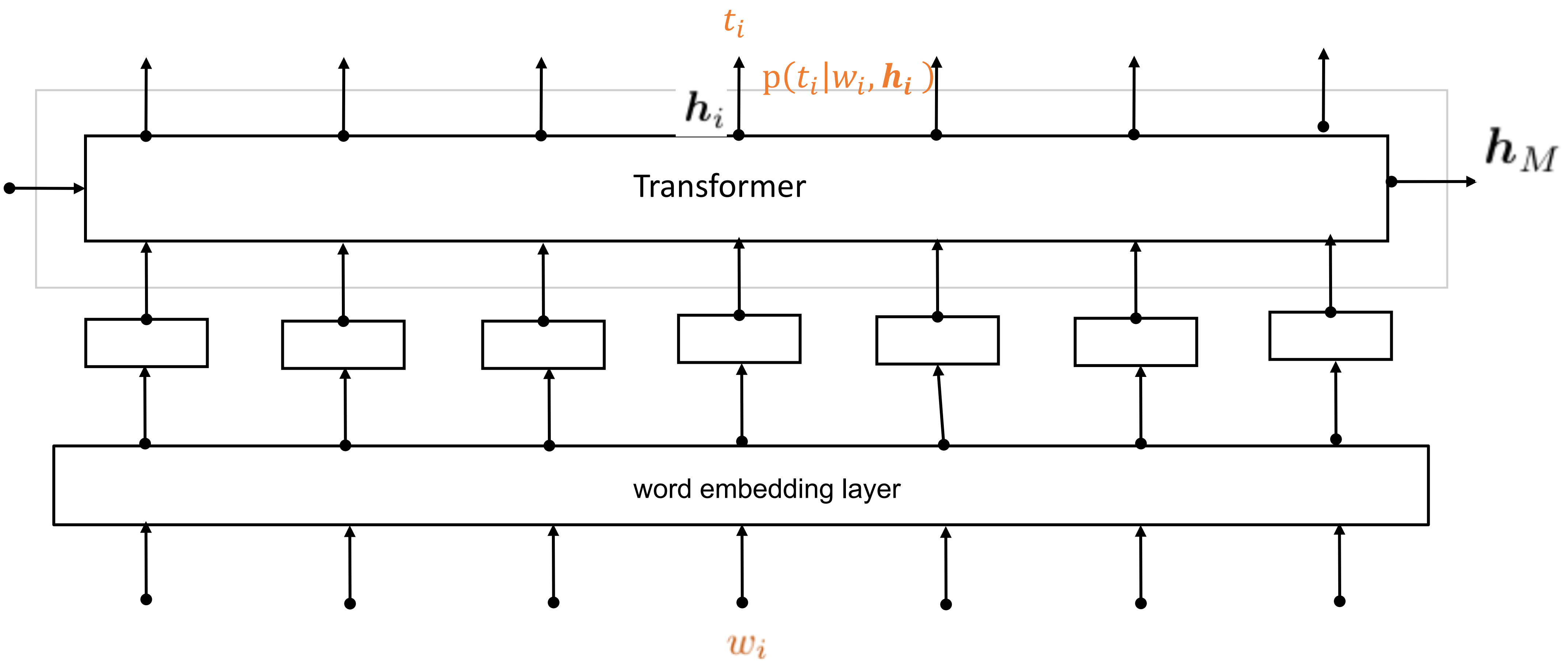
No Arrows / Directionality:
Each tag is dependent on all of it’s neighborhood

Possible Solution 3: Recurrent Neural Networks



$h_i \rightarrow$ contextual *vector* representations from previous words

Possible Solution 4: Transformers



$h_i \rightarrow$ contextual *vector* representations from **previous** and **future** words

Building POS Taggers

- **Supervised Learning:** Annotated corpora are used for training machine learning models, such as decision trees, SVMs, to predict POS tags for words.

- **Deep Learning:** Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Transformer-based models like BERT and GPT are used for sequence tagging tasks, including POS tagging.

- **Transfer Learning:** Pre-trained language models are fine-tuned for POS tagging tasks, leveraging their general language understanding to improve performance on specific tasks.

Chunking

- Chunking, involves identifying and grouping adjacent words or tokens in a sentence into meaningful units, often based on their grammatical structure or semantic roles.

1. **Noun Phrase (NP) Chunk:**
   - Example: "The big brown dog"
   - Chunk: "The big brown dog"
   - Information: This chunk represents a complete noun phrase consisting of a determiner ("The"), adjectives ("big" and "brown"), and a noun ("dog").
2. **Verb Phrase (VP) Chunk:**
   - Example: "ate lunch"
   - Chunk: "ate lunch"
   - Information: This chunk represents a verb phrase consisting of a verb ("ate") and a noun phrase ("lunch").
3. **Prepositional Phrase (PP) Chunk:**
   - Example: "in the park"
   - Chunk: "in the park"
   - Information: This chunk represents a prepositional phrase consisting of a preposition ("in") and a noun phrase ("the park").
4. **Named Entity (NE) Chunk:**
   - Example: "Apple Inc. is a tech company."
   - Chunk: "Apple Inc."

Rule Based Chunking

- Identify phrases from a sentence based on predefined grammatical rules and patterns.
- Typically relies on part-of-speech based patterns
- Example:
  - Extract all noun phrases from the sentence “The quick brown fox jumps over the lazy dog”
  - Noun phrases:
    - The quick brown fox
    - The lazy dog

Rule Based Chunking

- Define a POS RegEx based grammar rule

  r"NP: {<DT>?<JJ>*<NN.*>} ”

- Obtain part of speech tags for the input sentence first

  The_DT quick_JJ brown_JJ fox_NN jumps_VBZ over_IN the_DT lazy_JJ dog_NN ._.

- Extract token sequences that conform to the pattern.
  - The quick brown fox
  - the lazy dog

ML based Chunking

- Similar to PoS Taggeing, chunking involves solving sequence labeling problems.
- Approaches include, Supervised Machine Learning, Deep Learning and Transfer Learning based approaches.
- Tags are often specified by the BIO tagging scheme.

BIO tagging scheme for chunking

- **B: Beginning** - Indicates the first token of a chunk.
- **I: Inside** - Indicates a token inside a chunk (following the first token).
- **O: Outside** - Indicates a token that is not part of any chunk.

Example

- Sentence: "The quick brown fox jumps over the lazy dog.”
- Say, we are interested in Noun Chunking

The" - B-NP
"quick" - I-NP
"brown" - I-NP
"fox" - I-NP
"jumps" - O
"over" - O
"the" - B-NP
"lazy" - I-NP
"dog" - I-NP

Now:

Tutorial : Exploring POS and Chunkers

Next Week

- Deep Parsing and Information Extraction

Assignment 3: has been posted