Language Models for Law and Social Science

2. Tokenization

Tokenization: Overview

- Input:
  - A set of documents (e.g. text files), D.
- Output 1:
  - A sequence, W, containing a list of tokens – words or word pieces for use in natural language processing
- Output 2:
  - A document-term matrix, X, containing statistics about word/phrase frequencies in those documents.

Three Approaches to Tokenization

1. convert documents to count vectors, e.g. over pre-processed n-grams.
   - "bag of words" or "bag of terms" representation
   - used with topic models and classical ML
   - should be **informative/predictive** in the learning task, computationally **tractable**, and preferably somewhat **interpretable**.

Three Approaches to Tokenization

1. convert documents to count vectors, e.g. over pre-processed n-grams.
   - "bag of words" or "bag of terms" representation
   - used with topic models and classical ML
   - should be **informative/predictive** in the learning task, computationally **tractable**, and preferably somewhat **interpretable**.

2. segment documents into word pieces using byte pair encoding
   - maintain as much info from the original document as possible
   - for inputs to sequence models, i.e. transformers.

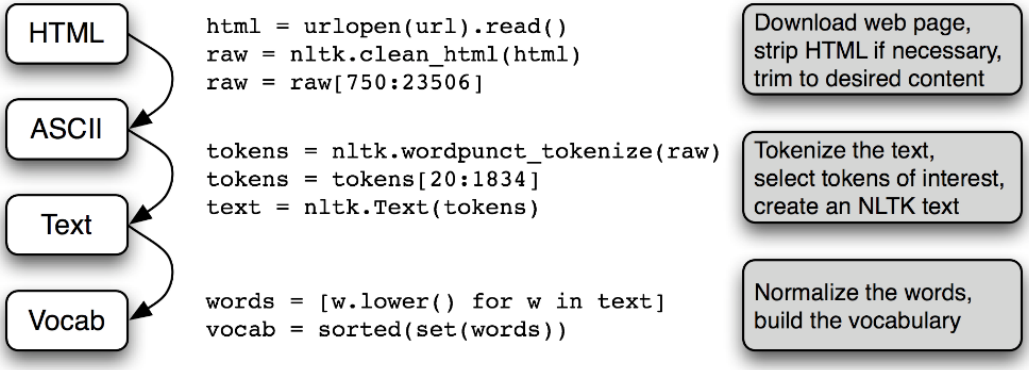
Three Approaches to Tokenization

1. convert documents to count vectors, e.g. over pre-processed n-grams.
   - "bag of words" or "bag of terms" representation
   - used with topic models and classical ML
   - should be **informative/predictive** in the learning task, computationally **tractable**, and preferably somewhat **interpretable**.

2. segment documents into word pieces using byte pair encoding
   - maintain as much info from the original document as possible
   - for inputs to sequence models, i.e. transformers.

3. enrich document with linguistics/grammar information
   - add more information to unprocessed doc based on sentence boundaries, parts of speech, syntax, etc
   - needed for specific tasks – eg relation extraction

A Standard Tokenization Pipeline



Source: NLTK Book, Chapter 3.

Pre-processing

- For many projects, the first question is: what data to throw out?
  - Uninformative data add noise and reduce statistical precision.
  - They are also computationally costly.
- Pre-processing choices can affect down-stream results, especially in unsupervised learning tasks (Denny and Spirling 2017).
  - in particular: some features are more interpretable, e.g. ("discretion", "have", "judge") vs ("the judge has discretion").

Normalizing Text

1. Capitalization
   - usually the capitalized/non-capitalized version of a word are equivalent → capitalization not informative.

Normalizing Text

1. Capitalization
   - usually the capitalized/non-capitalized version of a word are equivalent → capitalization not informative.
   - On the other hand: what about “the first amendment” versus “the First Amendment”?

Normalizing Text

1. Capitalization
   - usually the capitalized/non-capitalized version of a word are equivalent → capitalization not informative.
   - On the other hand: what about "the first amendment" versus "the First Amendment"?
2. Punctuation
   - the number of periods or commas in a document is usually not that useful
   - so in a bag of terms approach, punctuation can be dropped.

Normalizing Text

1. Capitalization
   - usually the capitalized/non-capitalized version of a word are equivalent → capitalization not informative.
   - On the other hand: what about "the first amendment" versus "the First Amendment"?
2. Punctuation
   - the number of periods or commas in a document is usually not that useful
   - so in a bag of terms approach, punctuation can be dropped.
   - but what about "Let's eat, Grandpa", versus "Let's eat Grandpa"?

Normalizing Text

1. Capitalization
   - usually the capitalized/non-capitalized version of a word are equivalent $\rightarrow$ capitalization not informative.
   - On the other hand: what about "the first amendment" versus "the First Amendment"?
2. Punctuation
   - the number of periods or commas in a document is usually not that useful
   - so in a bag of terms approach, punctuation can be dropped.
   - but what about "Let's eat, Grandpa", versus "Let's eat Grandpa"?
3. Numbers
   - individual numbers are usually too specific to keep in the vocabulary
   - But how often numbers are mentioned might be important; can replace with a special character, e.g. #.

Stopwords

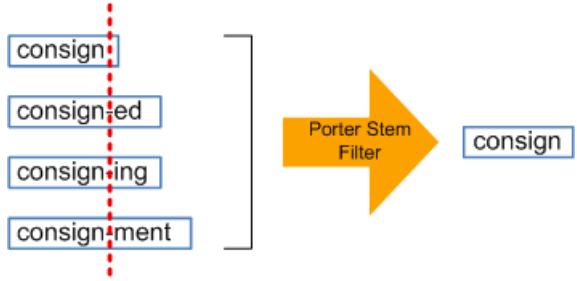
a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

Stopwords

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

- What about "not guilty"?
- Legal terms often contain stopwords:
  - "beyond a reasonable doubt"
  - "with all deliberate speed"
- can drop stopwords by themselves, but keep them when part of phrases.

Stemming/lemmatizing



- Effective dimension reduction with little loss of information.
- Lemmatizer produces real words, but N-grams won't make grammatical sense
  - e.g., "judges have been ruling" would become "judge have be rule"

Pre-processing with `gensim`

gensim.parsing.preprocessing.preprocess_string(*s*, *filters=[<function <lambda>>, <function strip_tags>, <function strip_punctuation>, <function strip_multiple_whitespaces>, <function strip_numeric>, <function remove_stopwords>, <function strip_short>, <function stem_text>]*)

Apply list of chosen filters to *s*.

Default list of filters:

- `strip_tags()`,
- `strip_punctuation()`,
- `strip_multiple_whitespaces()`,
- `strip_numeric()`,
- `remove_stopwords()`,
- `strip_short()`,
- `stem_text()`.

Parameters:
- **s** (*str*) –
- **filters** (*list of functions, optional*) –

Returns: Processed strings (cleaned).

Return type: list of str

Bag-of-words representation

Say we want to convert a corpus D to a matrix X:

- In the "bag-of-words" representation, a row of X is just the frequency distribution over words in the document corresponding to that row.

Counts and frequencies:

- **Document counts**: number of documents where a word appears.
- **Term counts**: number of total appearances of a word in corpus.
- **Term frequency**:

$$\text{Term Frequency of } w \text{ in document } k = \frac{\text{Count of } w \text{ in document } k}{\text{Total tokens in document } k}$$

Building a vocabulary

- What are the columns of the document-term matrix X?
  - Assign numerical indices to words to increase speed and reduce disk usage.
- Pick a number:, e.g. 100,000 most frequent words.

Building a vocabulary

- What are the columns of the document-term matrix X?
  - Assign numerical indices to words to increase speed and reduce disk usage.

- Pick a number:, e.g. 100,000 most frequent words.
- Frequency threshold:
  - Compute document frequencies for all words
  - Inspect low-frequency words and determine a minimum document threshold.
    - e.g., 10 documents, or .25% of documents.

scikit-learn's CountVectorizer

https://scikit-learn.org/stable/modules/feature_extraction.html

`CountVectorizer` implements both tokenization and occurrence counting in a single class:

```
>>> from sklearn.feature_extraction.text import CountVectorizer
```

This model has many parameters, however the default values are quite reasonable (please see the reference documentation for the details):

```
>>> vectorizer = CountVectorizer()
>>> vectorizer
CountVectorizer()
```

Let's use it to tokenize and count the word occurrences of a minimalistic corpus of text documents:

```
>>> corpus = [
...     'This is the first document.',
...     'This is the second second document.',
...     'And the third one.',
...     'Is this the first document?',
... ]
>>> X = vectorizer.fit_transform(corpus)
>>> X
<4x9 sparse matrix of type '<... 'numpy.int64'>'
    with 19 stored elements in Compressed Sparse ... format>
```

The default configuration tokenizes the string by extracting words of at least 2 letters.

- `corpus` is a sequence of strings, e.g. pandas data-frame columns.
- pre-processing options: strip accents, lowercase, drop stopwords,
- vocab options: min/max frequency, vocab size
- n-grams: can produce phrases up to length n (words or characters).

What about out-of-vocab words?

What about out-of-vocab words?

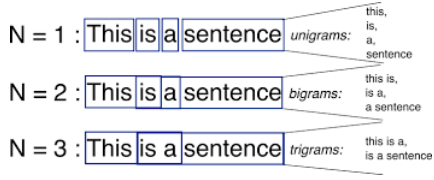
- in bag-of-words model:
  - drop them
  - replace with "unknown" token (<unk>)
  - replace with part-of-speech tag
  - replace with in-vocab hypernym (from WordNet)
  - others?

What about out-of-vocab words?

- in bag-of-words model:
  - drop them
  - replace with "unknown" token (<unk>)
  - replace with part-of-speech tag
  - replace with in-vocab hypernym (from WordNet)
  - others?
- alternative approaches that don't have this problem (more below):
  - hashing vectorizer
  - byte pair encoding

N-grams are phrases, sequences of words up to length N

- e.g. bigrams, trigrams, quadgrams, etc.



- Baseline for text classification of long documents (Google Developers Guide):
  - X = counts over bigrams, with vocab size = 20,000

`https://developers.google.com/machine-learning/guides/text-classification/step-3`

Feature selection

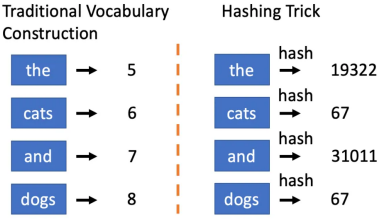
- N-grams quickly blow up the feature space.
  - filtering on frequency is easiest but not optimal – can filter on usefulness for a task instead.
- Text normalization is important (capitalization, punctuation, stopwords, stemming)

- For supervised learning tasks:
  - Use supervised feature selection to select predictive features (more on week 4)
- What about unsupervised learning (e.g. topic models)?
  - can use parts of speech / co-location statistics (week 3)

Feature selection

- N-grams quickly blow up the feature space.
  - filtering on frequency is easiest but not optimal – can filter on usefulness for a task instead.
- Text normalization is important (capitalization, punctuation, stopwords, stemming)
- For supervised learning tasks:
  - Use supervised feature selection to select predictive features (more on week 4)
- What about unsupervised learning (e.g. topic models)?
  - can use parts of speech / co-location statistics (week 3)
- In week 3, we explore more general problem of dimensionality reduction.

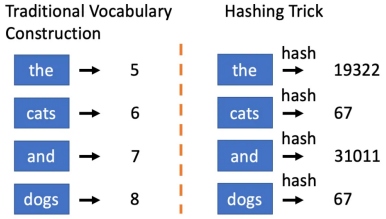
Hashing Vectorizer



- Rather than make a one-to-one lookup for each n-gram, put n-grams through a hashing function that takes an arbitrary string and deterministically outputs an integer in some range (e.g. 1 to 10,000).

```
>>> from sklearn.feature_extraction.text import HashingVectorizer
>>> hv = HashingVectorizer(n_features=10)
>>> hv.transform(corpus)
<4x10 sparse matrix of type '<... 'numpy.float64'>'
    with 16 stored elements in Compressed Sparse ... format>
```

Hashing Vectorizer



- Rather than make a one-to-one lookup for each n-gram, put n-grams through a hashing function that takes an arbitrary string and deterministically outputs an integer in some range (e.g. 1 to 10,000).

```
>>> from sklearn.feature_extraction.text import HashingVectorizer
>>> hv = HashingVectorizer(n_features=10)
>>> hv.transform(corpus)
<4x10 sparse matrix of type '<... 'numpy.float64'>'
    with 16 stored elements in Compressed Sparse ... format>
```

Pros:

- can have arbitrarilly small feature space
- handles out-of-vocabulary words – any word or n-gram gets assigned to an arbitrary integer based on the hash function.

Cons:

- harder to interpret features, at least not directly (eli5 implementation keeps track of the mapping)-
- collisions – n-grams will randomly be paired with each other in the feature map – in supervised learning, usually innocuous

Research Question

- What drives slant in print media in the United States?
  - **Consumer preferences?**
  - Politicians?
  - Newspaper owners?

- Example: What influences, whether a newspaper is more likely to use the term **death tax**, or the term **estate tax**?

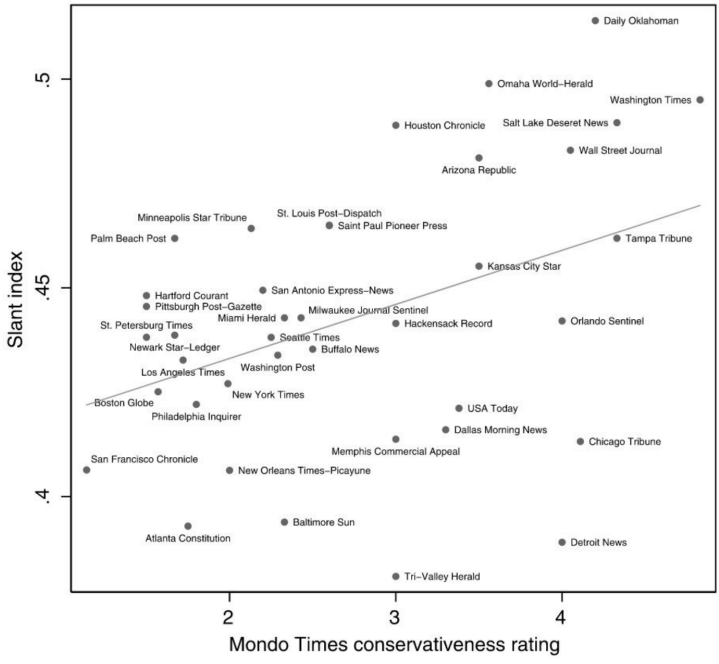
Corpus and data

- 2005 Congressional record and politicians' positions
  - https://www.congress.gov/congressional-record
    - Pre-processed to exclude common words
  - Vote share in the 2004 presidential election in politicians' constituencies
- Newspapers and their markets
  - newslibrary.com and proquest.com
    - Exclude opinion content
    - Exclude globally read newspapers (e.g. New York Times)
  - Owner information from E&P international yearbook (2000)
  - Demographic data from 2000 Census in the newspaper's PMSA (primary metropolitan statistical area)
  - Corporate political contributions (publicintegrity.org)

NLP Methods

$$\chi^2_{pl} = \frac{(f_{plr}f_{\sim pld} - f_{pld}f_{\sim plr})^2}{(f_{plr} + f_{pld})(f_{plr} + f_{\sim plr})(f_{pld} + f_{\sim pld})(f_{\sim plr} + f_{\sim pld})}.$$

- Pre-processing: remove common neutral words (the, if, …)
- Phrase selection
  - Select phrases which are used (a) frequently, and (b) significantly more by one party than the other.
  - Use 2- word phrases and 3-word phrases
  - 2x top 500 → 1000 phrases total
- Phrase to ideology mapping
  - Linear regression on congresspersons' ideologies
  - Use obtained slope parameters to estimate newspapers' ideologies
  - **0.4 correlation with Mondo times conservativeness rating of newspapers**
    - Could have repeated all following sections with this alternative rating


Daily Oklahoman
Omaha World-Herald
Washington Times
Houston Chronicle
Salt Lake Deseret News
Wall Street Journal
Arizona Republic
Minneapolis Star Tribune
St. Louis Post-Dispatch
Saint Paul Pioneer Press
Palm Beach Post
Tampa Tribune
Kansas City Star
Hartford Courant
San Antonio Express-News
Pittsburgh Post-Gazette
Miami Herald
Milwaukee Journal Sentinel
St. Petersburg Times
Hackensack Record
Orlando Sentinel
Seattle Times
Newark Star-Ledger
Buffalo News
Los Angeles Times
Washington Post
Boston Globe
New York Times
Philadelphia Inquirer
USA Today
Dallas Morning News
Chicago Tribune
Memphis Commercial Appeal
San Francisco Chronicle
New Orleans Times-Picayune
Baltimore Sun
Atlanta Constitution
Detroit News
Tri-Valley Herald
.5
.45
.4
Slant index
2
3
4
Mondo Times conservativeness rating

Panel A: Phrases Used More Often by Democrats

Two-Word Phrases

private accounts	Rosa Parks	workers rights
trade agreement	President budget	poor people
American people	Republican party	Republican leader
tax breaks	change the rules	Arctic refuge
trade deficit	minimum wage	cut funding
oil companies	budget deficit	American workers
credit card	Republican senators	living in poverty
nuclear option	privatization plan	Senate Republicans
war in Iraq	wildlife refuge	fuel efficiency
middle class	card companies	national wildlife

Three-Word Phrases

veterans health care	corporation for public broadcasting	cut health care
congressional black caucus		civil rights movement
VA health care	additional tax cuts	cuts to child support
billion in tax cuts	pay for tax cuts	drilling in the Arctic National
credit card companies	tax cuts for people	victims of gun violence
security trust fund	oil and gas companies	solvency of social security
social security trust	prescription drug bill	Voting Rights Act
privatize social security	caliber sniper rifles	war in Iraq and Afghanistan
American free trade	increase in the minimum wage	civil rights protections
central American free	system of checks and balances	credit card debt
	middle class families	

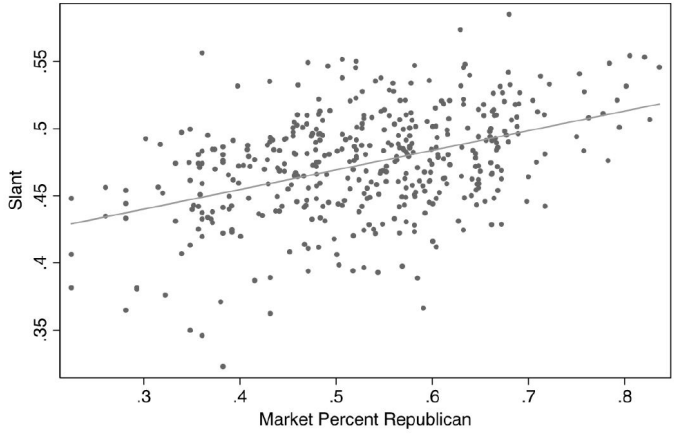
Panel B: Phrases Used More Often by Republicans

Two-Word Phrases

stem cell	personal accounts	retirement accounts
natural gas	Saddam Hussein	government spending
death tax	pass the bill	national forest
illegal aliens	private property	minority leader
class action	border security	urge support
war on terror	President announces	cell lines
embryonic stem	human life	cord blood
tax relief	Chief Justice	action lawsuits
illegal immigration	human embryos	economic growth
date the time	increase taxes	food program

Three-Word Phrases

embryonic stem cell	Circuit Court of Appeals	Tongass national forest
hate crimes legislation	death tax repeal	pluripotent stem cells
adult stem cells	housing and urban affairs	Supreme Court of Texas
oil for food program	million jobs created	Justice Priscilla Owen
personal retirement accounts	national flood insurance	Justice Janice Rogers
energy and natural resources	oil for food scandal	American Bar Association
global war on terror	private property rights	growth and job creation
hate crimes law	temporary worker program	natural gas natural
change hearts and minds	class action reform	Grand Ole Opry
global war on terrorism	Chief Justice Rehnquist	reform social security


Slant
.35
.4
.45
.5
.55
.3
.4
.5
.6
.7
.8
Market Percent Republican

Limitations of word tokenization

- modern language models are designed/intended to capture all meaning in texts.
  - with word tokenization, would need a massive vocabulary to capture all cases.
  - not impossible but computationally costly.
  - model might encounter new words in the test data.

Limitations of word tokenization

- modern language models are designed/intended to capture all meaning in texts.
  - with word tokenization, would need a massive vocabulary to capture all cases.
  - not impossible but computationally costly.
  - model might encounter new words in the test data.
- treats different word forms as separate words (e.g. “tax”, “taxes”, “taxed”)
  - or, with stemming, treats them as identical

Character tokenization

- alternative – tokenize characters rather than words:
  - "hello world" $\rightarrow$ {h,e,l,l,o, ,w,o,r,l,d}
  - by construction, no unkown words.

Character tokenization

- alternative – tokenize characters rather than words:
  - "hello world" $\rightarrow$ {h,e,l,l,o, ,w,o,r,l,d}
  - by construction, no unkown words.
- this actually works fine, and is used in some recent language models.

Character tokenization

- alternative – tokenize characters rather than words:
  - "hello world" $\rightarrow$ {h,e,l,l,o, ,w,o,r,l,d}
  - by construction, no unkown words.
- this actually works fine, and is used in some recent language models.
  - but not efficient: some single characters (e.g. "x") are much less frequent than some character subsequences (e.g. "tion") or even whole words (e.g. "the")

Subword Tokenization for Sequence Models

Most modern language models (e.g. BERT, GPT) use subword tokenization:

- construct character-level n-grams using byte pair encoding (frequent character sequences treated as a token)
- whitespace treated the same as letters
- all letters to lowercase, but add a special character for the next letter being capitalized.

Subword Tokenization for Sequence Models

Most modern language models (e.g. BERT, GPT) use subword tokenization:

- construct character-level n-grams using byte pair encoding (frequent character sequences treated as a token)
- whitespace treated the same as letters
- all letters to lowercase, but add a special character for the next letter being capitalized.

e.g., BERT's SentencePiece tokenizer:

- character-level byte-pair encoder, learns character n-grams to breaks words like "playing" into "play" and "##ing".
- have to fix a vocabulary size: e.g. BERT uses 30K.
- see notebook

Subword Tokenization for Sequence Models

Most modern language models (e.g. BERT, GPT) use subword tokenization:

- construct character-level n-grams using byte pair encoding (frequent character sequences treated as a token)
- whitespace treated the same as letters
- all letters to lowercase, but add a special character for the next letter being capitalized.

e.g., BERT's SentencePiece tokenizer:

- character-level byte-pair encoder, learns character n-grams to breaks words like "playing" into "play" and "##ing".
- have to fix a vocabulary size: e.g. BERT uses 30K.
- see notebook

GPT-3 tokenizer: `https://platform.openai.com/tokenizer`

Limitations of word/subword tokenization

- The tokenizers we have looked at so far are based only on data in the corpus.
  - bag-of-words tokenizers break down text into counts over the most relevant features.
  - subword tokenizers try to preserve the text as is.

Limitations of word/subword tokenization

- The tokenizers we have looked at so far are based only on data in the corpus.
  - bag-of-words tokenizers break down text into counts over the most relevant features.
  - subword tokenizers try to preserve the text as is.
- These tokenizers leave out a lot of information that we have from sophisticated and powerful conceptual models of language – that is, linguistics.

Segmenting paragraphs/sentences

- Many tasks should be done on sentences, rather than corpora as a whole.
  - spaCy is a good (but not perfect) job of splitting sentences, while accounting for periods on abbreviations, etc.
- There isn't a grammar-based paragraph tokenizer.
  - most corpora have new paragraphs annotated.
  - or use line breaks.

Parts of speech tags

- Parts of speech (POS) tags provide useful word categories corresponding to their functions in sentences:
  - Eight main parts of speech: verb (VB), noun (NN), pronoun (PR), adjective (JJ), adverb (RB), determinant (DT), preposition (IN), conjunction (CC).
  - The Penn TreeBank POS tag set (used in many applications) has 36 tags: `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`

Parts of speech tags

- Parts of speech (POS) tags provide useful word categories corresponding to their functions in sentences:
  - Eight main parts of speech: verb (VB), noun (NN), pronoun (PR), adjective (JJ), adverb (RB), determinant (DT), preposition (IN), conjunction (CC).
  - The Penn TreeBank POS tag set (used in many applications) has 36 tags: `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`
- Parts of speech vary in their informativeness for various functions:
  - For categorizing topics, nouns are usually most important
  - For sentiment, adjectives are usually most important.
- In particular, noun phrases are often informative features – spaCy can do fast noun phrase chunking.

Parts of speech tags

- Parts of speech (POS) tags provide useful word categories corresponding to their functions in sentences:
  - Eight main parts of speech: verb (VB), noun (NN), pronoun (PR), adjective (JJ), adverb (RB), determinant (DT), preposition (IN), conjunction (CC).
  - The Penn TreeBank POS tag set (used in many applications) has 36 tags: `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`
- Parts of speech vary in their informativeness for various functions:
  - For categorizing topics, nouns are usually most important
  - For sentiment, adjectives are usually most important.
- In particular, noun phrases are often informative features – spaCy can do fast noun phrase chunking.
- Can count POS tags as features – e.g., using more adjectives, or using more passive verbs.
  - provides style features, e.g. for authorship detection.

Named Entity Recognition

- Named entities such as “ETH Zurich” and “Marie Curie” are a special set of annotations, tagged by named entity recognizers (NER).
- usually identified by proper nouns; most pre-trained NER systems, like spACy, also give an entity category:

> [PER John Smith] , president of [ORG McCormik Industries] visited his niece [PER Paris] in [LOC Milan], reporters say .

Application: POS tags Predict Loan Repayment

Netzer, Lemaire, and Herzenstein (2019), "When Words Sweat"

Application: POS tags Predict Loan Repayment

Netzer, Lemaire, and Herzenstein (2019), "When Words Sweat"

- Imagine you consider lending $2,000 to one of two borrowers on a crowdfunding website. The borrowers are identical in terms of demographic and financial characteristics. However, the text they provided when applying for a loan differs:
  - Borrower #1: "*I am a hard working person, married for 25 years, and have two wonderful boys. Please let me explain why I need help. I would use the $2,000 loan to fix our roof. Thank you, god bless you, and I promise to pay you back.*"
  - Borrower #2: "*While the past year in our new place has been more than great, the roof is now leaking and I need to borrow $2,000 to cover the cost of the repair. I pay all bills (e.g., car loans, cable, utilities) on time.*"
- Which borrower is more likely to default?

Application: POS tags Predict Loan Repayment

Netzer, Lemaire, and Herzenstein (2019), "When Words Sweat"

- Imagine you consider lending $2,000 to one of two borrowers on a crowdfunding website. The borrowers are identical in terms of demographic and financial characteristics. However, the text they provided when applying for a loan differs:
  - Borrower #1: "*I am a hard working person, married for 25 years, and have two wonderful boys. Please let me explain why I need help. I would use the $2,000 loan to fix our roof. Thank you, god bless you, and I promise to pay you back.*"
  - Borrower #2: "*While the past year in our new place has been more than great, the roof is now leaking and I need to borrow $2,000 to cover the cost of the repair. I pay all bills (e.g., car loans, cable, utilities) on time.*"
- Which borrower is more likely to default?

"Loan requests written by defaulting borrowers are more likely to include words (or themes) related to the borrower's family, financial and general hardship, mentions of god, and the near future, as well as pleading lenders for help, and using verbs in present and future tenses."

Loan Application Words Predicting Repayment (Netzer, Lemaire, and Herzenstein 2019)

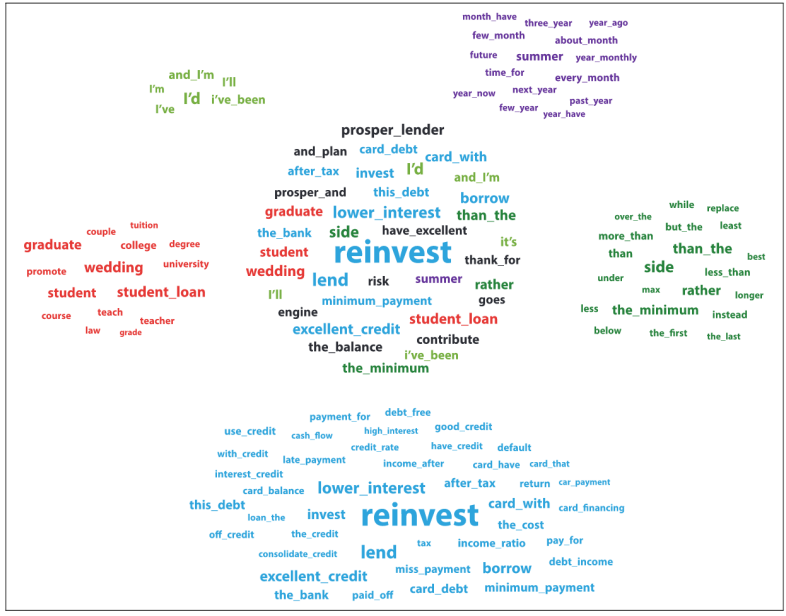


Figure 2. Words indicative of loan repayment.

Notes: The most common words appear in the middle cloud (cutoff = 1:1.5) and are then organized by themes. Starting on the right and moving clockwise: relative words, financial literacy words, words related to a brighter financial future, "I" words, and time-related words.

Loan Application Words Predicting Default (Netzer, Lemaire, and Herzenstein 2019)

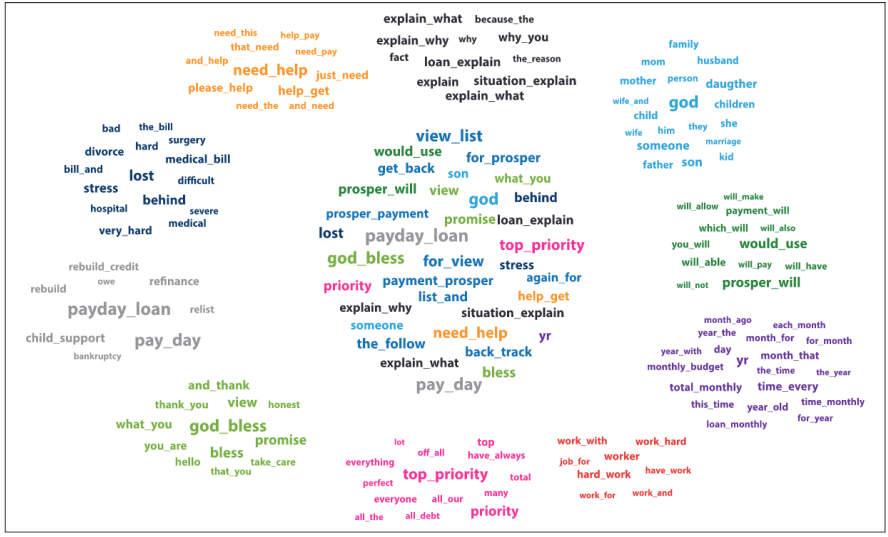


Figure 3. Words indicative of loan default.

Notes: The most common words appear in the middle cloud (cutoff = 1:1.5) and are then organized by themes. Starting on the top and moving clockwise: words related to explanations, external influence words and others, future-tense words, time-related words, work-related words, extremity words, words appealing to lenders, words relating to financial hardship, words relating to general hardship, and desperation/plea words.

Using Grammar: Constituency

Using Grammar: Constituency

- The idea of constituency is that groups of words behave as singular functional units in a sentence.
- Some example noun phrases:

Harry the Horse	a high-class spot such as Mindy's
the Broadway coppers	the reason he comes into the Hot Box
they	three parties from Brooklyn

  - these phrases consist of many POS's but function as nouns

Using Grammar: Constituency

- The idea of constituency is that groups of words behave as singular functional units in a sentence.
- Some example noun phrases:

Harry the Horse	a high-class spot such as Mindy's
the Broadway coppers	the reason he comes into the Hot Box
they	three parties from Brooklyn

  - these phrases consist of many POS's but function as nouns
- In English, constituents can be moved around in a sentence (e.g. these prepositional phrases):
  - John talked [to the students] [about linguistics].
  - John talked [about linguistics] [to the students] .

Using Grammar: Syntactic Dependencies

- The basic idea:
  - **Syntactic structure** consists of **words**, linked by binary directed relations called **dependencies**.
  - Dependencies identify the grammatical relations between words.

Dependencies: Binary Directed Relations Between Words (Head and Dependent)

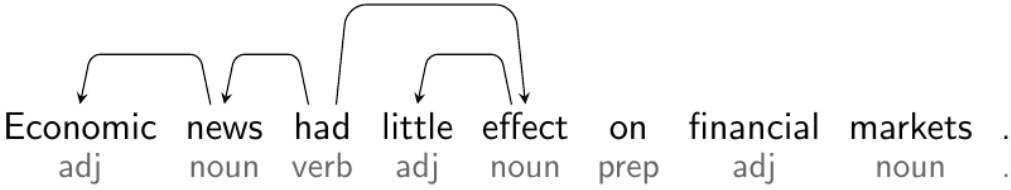
Economic	news	had	little	effect	on	financial	markets	.
adj	noun	verb	adj	noun	prep	adj	noun	.

- dependency trees are mostly determined by the ordering of POS tags.

Dependencies: Binary Directed Relations Between Words (Head and Dependent)

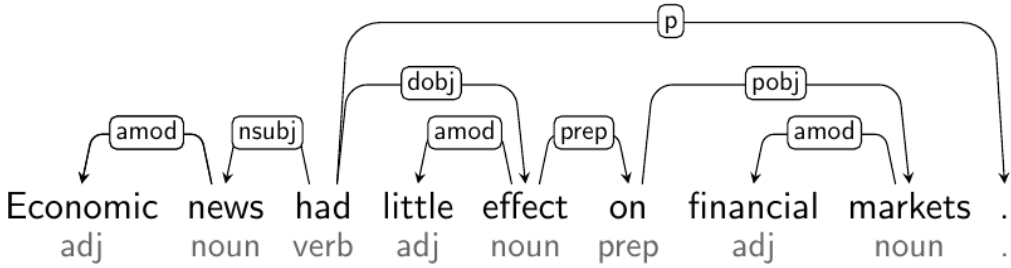
- the "root" of a sentence is the main verb (for compound sentences, the first verb).

Dependencies: Binary Directed Relations Between Words (Head and Dependent)



- directed arcs indicate dependencies: a one-way link from a "head" token to a "dependent" token.
- A word can be "head" multiple times, but "dependent" only once.

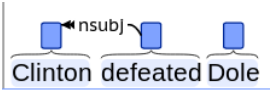
Dependencies: Binary Directed Relations Between Words (Head and Dependent)



- arc labels indicate functional relations, e.g.:
  - nsubj: verb → subject doing the verb
  - dobj: verb → object targeted by the verb
  - amod: noun → attribute of the noun
- spaCy dependency visualizer: `https://explosion.ai/demos/displacy`

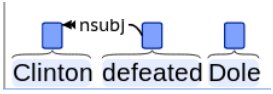
Who does What to Whom: Subjects

- **nsubj: nominal subject**
  - points from the active verb to the agent subject.

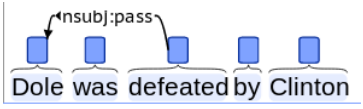


Who does What to Whom: Subjects

- **nsubj: nominal subject**
  - points from the active verb to the agent subject.



- **nsubjpass: passive nominal subject**
  - points from a passive veb to the patient subject



Who does What to Whom: Objects

dobj: direct object

- points from an active verb to the the (accusative) object noun phrase.

"She **gave** me a **raise**"

$$gave \overset{dobj}{\rightarrow} raise$$

Who does What to Whom: Objects

dobj: direct object

- points from an active verb to the the (accusative) object noun phrase.

"She **gave** me a **raise**"

$$gave \overset{dobj}{\rightarrow} raise$$

dative: dative or indirect object

- points from an active verb to the the (dative) object noun phrase.

"She **gave me** a raise"

$$gave \overset{dative}{\rightarrow} me$$

Who does What to Whom: Objects

dobj: direct object

- points from an active verb to the the (accusative) object noun phrase.

"She **gave** me a **raise**"

$$gave \overset{dobj}{\rightarrow} raise$$

dative: dative or indirect object

- points from an active verb to the the (dative) object noun phrase.

"She **gave me** a raise"

$$gave \overset{dative}{\rightarrow} me$$

pobj: object of a preposition

- noun phrase following a preposition

"I sat **on** the **chair**"

$$on \overset{pobj}{\rightarrow} chair$$

What Attributes do Entities Have?

acomp: adjectival complement

- points from verb to adjectival phrase functioning as object

"Bill **is honest** ": acomp(is $\rightarrow$ honest)

What Attributes do Entities Have?

acomp: adjectival complement

- points from verb to adjectival phrase functioning as object

"Bill **is honest** ": acomp(is $\rightarrow$ honest)

attr: attribute

- points from copula verb to an attribute noun phrase.

"Bill **is** a **saint**": attr(is $\rightarrow$ saint)

What Attributes do Entities Have?

acomp: adjectival complement

- points from verb to adjectival phrase functioning as object

"Bill **is honest** ": acomp(is $\rightarrow$ honest)

attr: attribute

- points from copula verb to an attribute noun phrase.

"Bill **is** a **saint**": attr(is $\rightarrow$ saint)

amod: adjectival modifier

- points from a noun to an adjective modifying it

"Sam eats **red meat**": amod(meat $\rightarrow$ red)

Verb phrases

- **aux: auxiliary**
  - points from a main verb to a helping verb, including modals.

"Reagan **has died**": aux(died → has)

"He **should leave**": aux(leave → should)

Verb phrases

- **aux: auxiliary**
  - points from a main verb to a helping verb, including modals.

"Reagan **has died**": aux(died → has)

"He **should leave**": aux(leave → should)

- **auxpass: passive auxiliary**
  - points from a main verb to a helping verb indicative passive voice.

"Laws have **been broken**": auxpass(broken → been)

Verb phrases

- **aux: auxiliary**
  - points from a main verb to a helping verb, including modals.

    "Reagan **has died**": aux(died $\rightarrow$ has)

    "He **should leave**": aux(leave $\rightarrow$ should)
- **auxpass: passive auxiliary**
  - points from a main verb to a helping verb indicative passive voice.

    "Laws have **been broken**": auxpass(broken $\rightarrow$ been)
- **neg: negation modifier**
  - points from a verb to a negation indicator
  - "Bill **is not** a scientist": neg(is $\rightarrow$ not)

Verb phrases

- **aux: auxiliary**
  - points from a main verb to a helping verb, including modals.

    "Reagan **has died**": aux(died → has)

    "He **should leave**": aux(leave → should)
- **auxpass: passive auxiliary**
  - points from a main verb to a helping verb indicative passive voice.

    "Laws have **been broken**": auxpass(broken → been)
- **neg: negation modifier**
  - points from a verb to a negation indicator
  - "Bill **is not** a scientist": neg(is → not)
- **prt: phrasal verb particle**
  - points from a verb to its particle, linking phrasal verbs.

    "They **shut down** the station": prt(shut → down)

Verb phrases

- **aux: auxiliary**
  - points from a main verb to a helping verb, including modals.

    "Reagan **has died**": aux(died → has)

    "He **should leave**": aux(leave → should)
- **auxpass: passive auxiliary**
  - points from a main verb to a helping verb indicative passive voice.

    "Laws have **been broken**": auxpass(broken → been)
- **neg: negation modifier**
  - points from a verb to a negation indicator
  - "Bill **is not** a scientist": neg(is → not)
- **prt: phrasal verb particle**
  - points from a verb to its particle, linking phrasal verbs.

    "They **shut down** the station": prt(shut → down)
- and more...

Semantic Role Labeling (PropBank Labels)

Ex1: [$_{\text{Arg0}}$ The group] *agreed* [$_{\text{Arg1}}$ it wouldn't make an offer].

Ex2: [$_{\text{ArgM-TMP}}$ Usually] [$_{\text{Arg0}}$ John] *agrees* [$_{\text{Arg2}}$ with Mary] [$_{\text{Arg1}}$ on everything].

ARG0	agent	ARG3	starting point, benefactive, attribute
ARG1	patient	ARG4	ending point
ARG2	instrument, benefactive, attribute	ARGM	modifier

Table 1.1: List of arguments in PropBank

- Agent (ARG0)
  - Volitional/sentient involvement in event or state
  - Causes an event or change of state in another participant
- Patient (ARG1)
  - Causally affected by an agent/action
  - Undergoes change of state
- ARG2 has three functions:
  - instrument for an action ("Pat opened the door with a crowbar.")
  - attribute assigned to a patient ("Pat is an agent").
  - benefactive: the dative/indirect object ("Sasha gave the crowbar to Pat.")

ARG-M: Modifiers

ArgM-TMP	when?	yesterday evening, now
LOC	where?	at the museum, in San Francisco
DIR	where to/from?	down, to Bangkok
MNR	how?	clearly, with much enthusiasm
PRP/CAU	why?	because ... , in response to the ruling
REC		themselves, each other
ADV	miscellaneous	
PRD	secondary predication	...ate the meat raw

- AllenNLP semantic role labeling demo:

  `https://demo.allennlp.org/semantic-role-labeling`

: Ash, Jacobs, MacLeod, Naidu, and Stammbach (2020) "Unsupervised extraction of rights and duties from collective bargaining agreements"

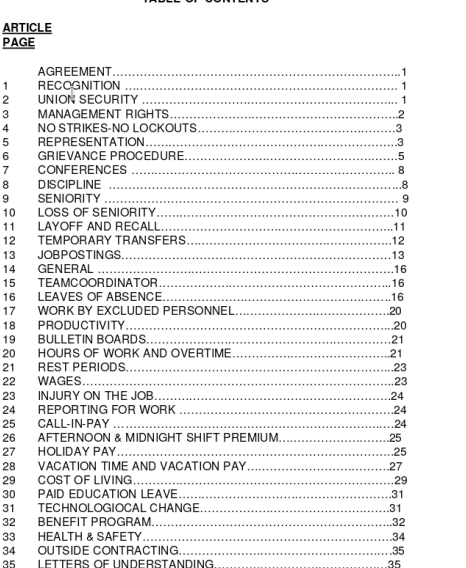


TABLE OF CONTENTS

ARTICLE PAGE

AGREEMENT

This Agreement ratified December 16,2005 is made and entered into between ST. CLAIR TECHNOLOGIES INC., Wallaceburg, Ontario (hereinafter called "the Company"), and the International Union, united Automobile, Aerospace and Agricultural Implement Workers of America (UAW-CLC) and its Local No. 251, (hereinafter called "the Union").

ARTICLE 1
RECOGNITION

1. The provisions of this Agreement shall apply to all employees covered by this Agreement without discrimination on account of race, creed, colour, sex, marital status, nationality, ancestry or place of origin.

2. Wherever the male noun or pronoun is used, it shall also mean the female.

3. The Company recognizes the Union as the sole bargaining agent of all its employees at Wallaceburg, Ontario, save and except supervisor, those above the rank of supervisor, office and sales staff, students for not more than twenty four hours per week and students employed during the school vacation period (May 1st-September 15th). In case of reduction in force, students would be laid off first. Students will be paid at a rate to be determined by the Company, but will not be less than the Employment Standards Act.

4. The word "employee" or "employees" wherever used in this Agreement shall mean only the employees in the bargaining unit defined above unless the context otherwise provides.

5. The Company will negotiate with the Union for the purpose of adjusting any disputes which may arise concerning sickness and accident, wages, hours and working conditions.

ARTICLE 2

2. Hire, promote, demote, classify, transfer, suspend and retire employees, and to discipline or discharge for just cause, any employee provided that a claim by an employee who has acquired seniority that he has been discharged or disciplined without just cause may be the subject of a grievance and dealt with as here in before provided.

3. Make, enforce, and alter, from time to time, rules and regulations to be observed by the employees, such rules not to be inconsistent with the provisions of this Agreement. The Company agrees to give a copy of any changes in plant rules to the Union Chairperson prior to posting of same on bulletin boards.

4. Determine the nature and kind of business conducted by the Company, the kinds and locations of plants, equipment and material to be used, the control of materials and parts, the use of incentive programs, the methods and techniques of work, the content of jobs, the schedules of production, the number of employees to be employed, the extension, limitations, curtailment or cessation of operations or any part thereof, and to determine and exercise all other function and prerogatives which shall remain solely with the Company except specifically limited by the express provisions of this Agreement.

ARTICLE 4
NO STRIKES - NO LOCKOUTS

1. The Union agrees that during the term of this agreement, there shall be no strikes, sit-downs, work stoppage, slowdowns, or suspension of work, either complete or partial, for any reason, by an employee or employees. There shall be no lockout of employees by the Company, for the duration of this Agreement.

ARTICLE 5
REPRESENTATION

UNION SECURITY

1. All employees covered by this Agreement who are members of the Union at the signing date of this Agreement or who after become members thereof during the term of this Agreement, must retain their membership in the Union for the duration of the Agreement by paying the regular monthly dues levied against all members, as a condition of employment. All employees covered by this Agreement who are not members of the Union shall pay regular monthly dues, the same as the dues that are levied against those who are members of the union as a condition of employment.

2. All new employees, upon completion of thirty (30) days employment shall become members thereof in good standing in accordance with the constitution and bylaws of the Union for the life of this Agreement.

3. The Company will during the term of the Agreement, deduct initiation fees, monthly dues and assessments on a monthly basis from the pay cheque of all seniority employees, probationary employees and full-time students who have worked or been compensated for forty (40) hours in any one (1) month, or as required by the U.A.W. constitution, (full-time student being a student who works all or any time between May 1st and September 15th of the same year). Such deductions shall be credited to the Secretary-Treasurer of Local 251, not later than the tenth (10th) day of the calendar month next following the month in which such deductions are made. The Company and the Union will work out a mutually satisfactory arrangement by which the Company will furnish monthly records to the Financial Secretary of Local 251 of those from whom deductions were made, together with the amount of such deductions.

ARTICLE 3
MANAGEMENT RIGHTS

The Union recognizes and acknowledges that the management of the plant and direction of the working force are fixed exclusively in the Company and, without restricting the generality of the foregoing, the Union acknowledges that it is the exclusive function of the Company to:

1. Maintain order and efficiency

4

appointed by the Union and work on the afternoon or midnight shift during such periods as the Company schedules these shifts and is equal to or greater than five (5) employees. Stewards will have preferred seniority on their shift for lay off and recall purposes only.

3. The Union will inform the Company in writing of the names of the stewards and members of the Grievance Committee and of any subsequent changes in the names of any steward or members of the Grievance Committee. The Company shall not be asked to recognize any steward or member of the Grievance Committee until such notification from the Union has been received.

4. The Union acknowledges that committee persons and stewards have their regular duties as employees to perform and that such persons will not leave their regular duties without first obtaining permission from their supervisor. Such permission shall not be unreasonably withheld. In the application of representation language "such permission shall not be unreasonably withheld" it is understood that a supervisor will not detain a Union representative from performing legitimate representation and by the same token the Union representative will understand the occasional need to complete a job in the interest of continuing production before leaving for legitimate union business. In any event, no such Union representative shall be detained any longer than thirty (30) minutes to perform their union representation duties.

5. The Company shall schedule a meeting, date and time within the time limits prescribed for any grievance submitted to Step 2 and/or Step 3 of the grievance procedure. The grievance committee only shall be compensated at their job rate for any regular scheduled work hours lost during such meeting with Company representatives. Overtime shall be paid when the meeting has been requested by the Company or the meeting goes beyond the Union representatives scheduled shift.

Text Pre-Processing Steps

- Contracts arrived as PDFs, along with matched metadata.
- Convert PDFs to machine-readable text (best was ABBBY FineReader)

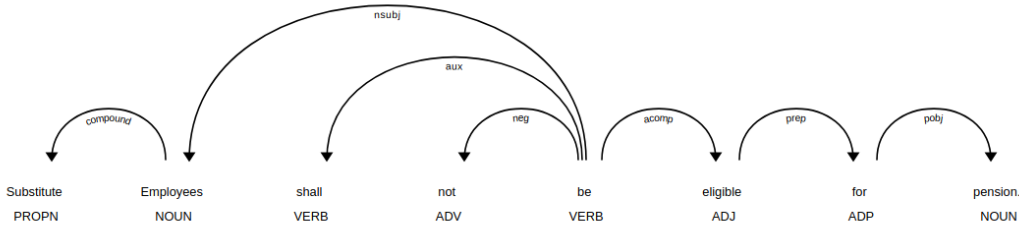
Text Pre-Processing Steps

- Contracts arrived as PDFs, along with matched metadata.
- Convert PDFs to machine-readable text (best was ABBBY FineReader)
- Exclude text for wage schedules, exhibits, appendices, etc.

Text Pre-Processing Steps

- Contracts arrived as PDFs, along with matched metadata.
- Convert PDFs to machine-readable text (best was ABBBY FineReader)
- Exclude text for wage schedules, exhibits, appendices, etc.
- Split the contracts into sections (RegEx) and sentences (spaCy):
  - 980,909 contract sections (33 per contract), 10.8 million sentences (11 per section)
- Co-reference resolution by section: replace pronouns with referent entity

Syntactic Parse for Contract Statements



- Dependency parsing (spaCy):
  - Output: Parse tree, giving functional relations between words in a sentence.
  - Identify syntactic subjects, and form statements around each subject
- Pipeline extracts clauses of the form: **Subject, Verb, Object**

Parse Information on Subjects and Verbs

- Subject categories:
  - worker, firm, union, manager

Parse Information on Subjects and Verbs

- Subject categories:
  - worker, firm, union, manager
- Deontic modal verbs (deontic indicating "duty") capture necessity/possibility in social freedoms to act:
  - strict (*shall*, *will*, *must*) modals express necessity
  - permissive (*may*, *can*) modals express possibility

Parse Information on Subjects and Verbs

- Subject categories:
  - worker, firm, union, manager
- Deontic modal verbs (deontic indicating "duty") capture necessity/possibility in social freedoms to act:
  - strict (*shall*, *will*, *must*) modals express necessity
  - permissive (*may*, *can*) modals express possibility
- Parser indicates negation ("shall **not**") and active/passive ("shall provide" vs "shall be provided")

Parse Information on Subjects and Verbs

- Subject categories:
  - worker, firm, union, manager
- Deontic modal verbs (deontic indicating "duty") capture necessity/possibility in social freedoms to act:
  - strict (*shall*, *will*, *must*) modals express necessity
  - permissive (*may*, *can*) modals express possibility
- Parser indicates negation ("shall **not**") and active/passive ("shall provide" vs "shall be provided")
- Special verbs:
  - *Obligation* Verbs (have to, ought to, be required, be expected, be compelled, be obliged, be obligated)
  - *Prohibition* Verbs (be prohibited, be forbidden, be banned, be barred, be restricted, be proscribed)
  - *Permission* Verbs (be allowed, be permitted, be authorized)
  - *Entitlement* Verbs (have, receive, retain)

Contract Statement Typology (Simplified)

Based on human (lawyer) annotation, machine assignments have precision of 91-99%
(Ash, Jacobs, MacLeod, Naidu, Stammbach 2020).

Categorization Logic	Examples
Obligations	
Positive & Strict Modal & Active Verb	shall provide, shall include, shall notify, shall continue
Positive & Strict Modal & Obligation Verb	shall be required, shall be expected, shall be obliged
Positive & Non-Modal & Obligation Verb	is required, is expected
Prohibitions	
Negative & Any Modal & Active Verb	shall not exceed, shall not use, shall not apply
Negative & Permission Verb	shall not be allowed, is not permitted
Positive & Strict Modal & Constraint Verb	shall be prohibited, shall be restricted
Permissions	
Positive & Non-Modal & Permission Verb	is allowed, is permitted, is authorized
Positive & Strict Modal & Permission Verb	shall be allowed, shall be permitted
Positive & Permissive Modal & Active Verb	may be, may request, may use, may require, may apply
Negative & Any Modal & Constraint Verb	shall not be restricted, shall not be prohibited
Rights	
Strict Modal & Passive Verb	shall be paid, shall be given, shall not be discharged
Positive & Strict Modal & Entitlement Verb	shall have, shall receive, shall retain
Negative & Any Modal & Obligation Verb	may not be required

Summary Stats: Statement Type Shares

Subject	Clause Type				
	Obligation (%)	Prohibition (%)	Permission (%)	Right (%)	*Total (%)*
Worker	20.9	3.1	8.4	22.9	*55.3*
Firm	24.7	1.5	3.4	0.9	*30.5*
Union	7.0	0.6	2.0	2.1	*11.7*
Manager	1.7	0.1	0.4	0.2	*2.5*
Total	*54.4*	*5.3*	*14.1*	*26.2*	*100.0*

- Contracts consist mostly of worker rights (22.9%), worker obligations (20.9%) and firm obligations (24.7%)
- Firm rights are rare (0.9%); makes sense as management reserves rights.