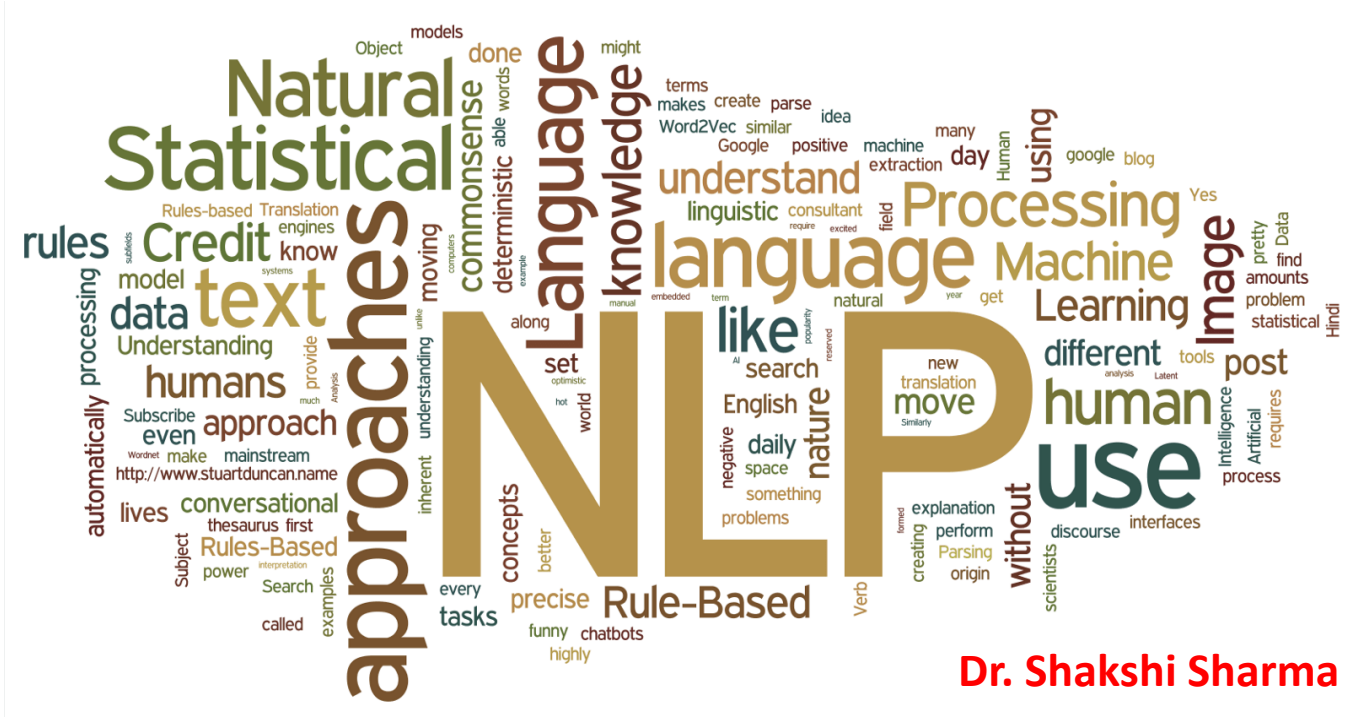
Natural Language Processing



Dr. Shakshi Sharma

Assistant Professor

School of Artificial Intelligence (SoAI), Bennett University

PhD|Junior Researcher|Estonia (Europe)

Research Intern|University of Sheffield, United Kingdom (UK)

Website: https://sites.google.com/view/shakshi-sharma/home

Coreference Resolution in NLP

What is Coreference Resolution?

- Coreference Resolution comes with NLP that tries to find all noun phrases refer to the same real-world entity.
- Suppose you have to find the pronouns in a sentence and replace them with relevant nouns. Coreference resolution can be used to do that.
- It finds and groups the words which refer to the same entities and replaces pronouns with noun phrases.

What is Coreference Resolution?

Consider the following sentence.

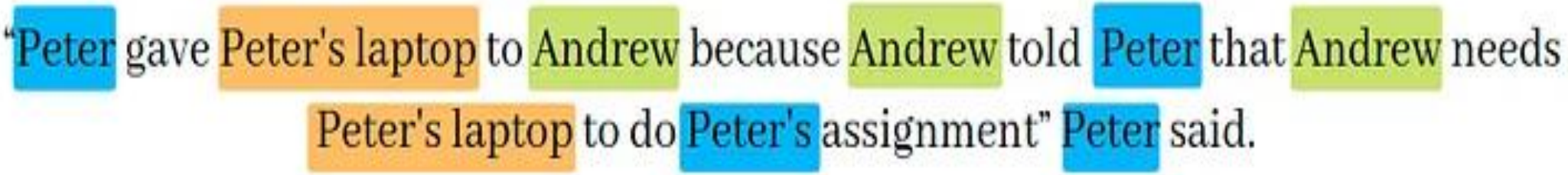
“I gave my laptop to Andrew because he told me that he needs it to do his assignment” Peter said.

- In Coreference Resolution first, it **groups** the words into several groups by considering **entities**. In this sentence the main entities are
- Andrew
- Peter
- Peter’s Laptop

What is Coreference Resolution?

- According to those entities, it can divide nouns and pronouns into

After that, it replaces all the pronouns in the sentence with relevant nouns.



What is Coreference Resolution?

Consider the following sentence.

E.g.1 **Narendra Modi, the prime minister of India communicate with common people through ‘Man ki baat’ program.**

E.g.2 **The music was so loud that it could not be enjoyed.**

E.g. 3 **Despite her difficulty, Swati went ahead to help him**

Types of Coreference

- **Anaphora:** Refers to when a pronoun or other referring expression points back to a previous mention.

Types of Coreference

- **Anaphora:** Refers to when a pronoun or other referring expression points back to a previous mention.

For example, "**Alice bought a new phone. She really likes it. The device has great features**."

- In this example, the following anaphoric references occur:
- "**She**": This pronoun refers to "**Alice**", who is the person mentioned in the first sentence.
- "**It**": This pronoun refers to "**phone**", which is mentioned in the first sentence as well.
- "**The device**": This noun phrase also refers to "**phone**", continuing the reference from the first sentence.

Types of Coreference

➢**Cataphora:**

Refers to when a reference is made to a future mention.

For example, in the sentence "**Before she went to the store, Alice had already decided what to buy.**"

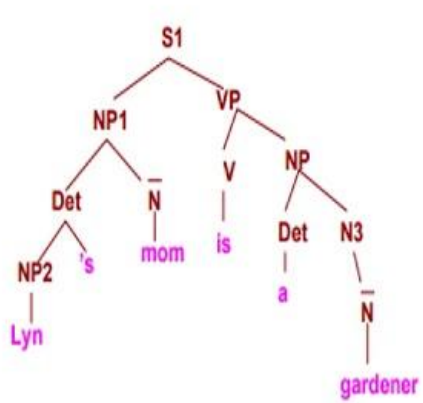
"**she**" refers to "**Alice**" which is mentioned later.

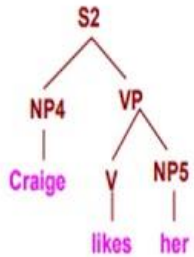
Way to build a reference resolution system

- Rule based (Hobbs Algorithm)
- Supervised algorithms (it’s a classification task)

Hobb's Algorithm

- Simple syntax-based algorithm that on syntactic parser .
- Searches syntactic trees of current and preceding sentences in breadth-first, left-to-right manner.
- Stops when it finds matching NP.





- Start search at NP5 in S2.
- Reject NP4 as no NP node between it and X (S2).
- What would have happened if the subject was *Craige's mom*?
- Move to S1. NP1 is first NP we encounter, so finish.
- Result: *Lyn's mom*

Implementing coreference in Python

- **Neural Coref library** in **Spacy** can be used

- **Hobbs** algorithm code available in **Github**

All **anaphoric** relations are not **coreferential** or vice versa.

E.g. We went to see a movie last night. The tickets were so expensive.

E.g. Narendra Modi or Modi is the prime minister of India.

When do we use Coreference Resolution?

- Coreference resolution is using in a variety of NLP tasks such as,

- Text summarization
- Information extraction (Named entity extraction)
- Sentiment analysis
- Machine translation
- Information Retrieval

Contextual Words and Phrases

Contextual words and phrases refer to words or phrases whose meaning **depends on the surrounding text or context**. Understanding context is crucial for accurate language processing.

Contextual Words and Phrases

Contextual words and phrases refer to words or phrases whose meaning **depends on the surrounding text or context**. Understanding context is crucial for accurate language processing.

Examples:

- **Word Sense Disambiguation**: In the sentences "**He went to the bank to fish**" and "**He deposited money at the bank**," understanding that "**bank**" refers to a **riverbank** in the first and a **financial institution** in the second is achieved through context.

- **Phrase Understanding**: In the phrase "**the glass broke**," the meaning of "**glass**" (as in a drinking container) is understood based on the context of "broke."

Importance of Context

- **Meaning Disambiguation**: **Words** can have different meanings based on context. For example, "**bank**" can refer to a financial institution or the side of a river. Understanding the surrounding text helps determine the correct meaning.

- **Coherence and Flow**: **Context** helps maintain the coherence and flow of sentences and paragraphs, allowing for a more natural understanding of the text.

Contextual Models:

- **Word Embeddings**: Traditional embeddings like **Word2Vec** or **GloVe** provide a static representation of words. Contextual embeddings go beyond this by considering the surrounding text.

- **Transformers**: Models like **BERT** (*Bidirectional Encoder Representations from Transformers*) and **GPT** (*Generative Pre-trained Transformer*) generate dynamic, context-sensitive representations of words.

For example, BERT captures the meaning of a word in the context of its sentence, while GPT generates contextually appropriate continuations of text.

Homonyms in NLP

Homonyms

- Homonyms are words that are spelled or pronounced the same but have **different meanings.**

- They can be a **challenge** for NLP systems because they require distinguishing between different meanings based on context.

Types of Homonyms:

- **Homophones**: Words that **sound the same** but have different meanings (e.g., "to," "too," and "two").

- **Homographs**: Words that are **spelled the same** but have different meanings and possibly different pronunciations (e.g., "lead" as in to guide vs. "lead" as in the metal).

Challenges

- **Disambiguation**: Identifying the correct meaning of a homonym requires understanding the context in which it is used.

For instance, distinguishing between "lead" (to guide) and "lead" (the metal) based on their usage in different sentences.

Handling Homonyms in NLP:

- **Contextual Embeddings**: Modern models like **BERT** and **GPT** handle homonyms effectively by providing contextually relevant meanings. For example, BERT's *attention* mechanism helps disambiguate homonyms by considering surrounding words and phrases.

- **Word Sense Disambiguation (WSD)**: Algorithms designed to determine the correct sense of a word based on its context. These can be *rule-based, statistical, or machine learning-based.*

Applications and Implications of Homonyms

- **Improved Communication**: Accurate contextual understanding and homonym disambiguation are crucial for applications like **machine translation**, where misinterpreting homonyms can lead to **incorrect** translations.

- **Enhanced Search and Retrieval**: Contextual understanding improves **search engines** and information retrieval systems by returning more relevant results based on the nuanced meaning of queries.

- **Natural Language Understanding**: Better handling of context and homonyms leads to more sophisticated **conversational agents and virtual assistants**, improving their ability to understand and generate human-like responses.

Synsets, Hypernyms in NLP

Synsets

- **Definition**: Synsets (short for "synonym sets") are groups of words or phrases that are considered **synonymous** in a particular context. **Each synset represents a distinct concept or idea.**

- **Purpose**: Synsets are used to capture and represent the semantic relationships between words. They help in understanding that different words or phrases can convey similar meanings.

Example:

In **WordNet**, a large lexical database of English, the word "**car**" might be grouped with "**automobile**," "**motorcar**," and "**machine**" in a synset. These words are considered synonyms within the context of the concept of a vehicle.

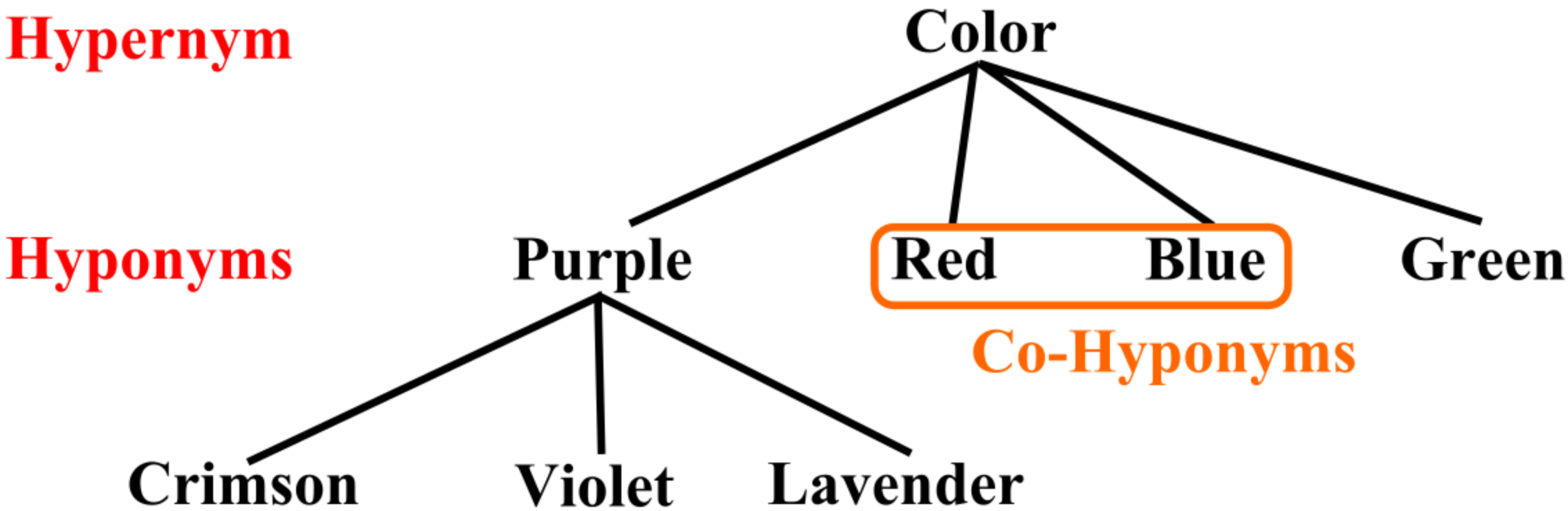
Hypernyms

- **Definition**: Hypernyms are words that denote a **broader category or a more general term in relation to other words**. They represent a higher-level concept under which more specific terms (hyponyms) fall.

- **Purpose**: Hypernyms help in organizing words into a hierarchical structure, which is useful for understanding the relationship between different concepts and for tasks like word sense disambiguation and semantic search.

Example:

For the synset containing "**car**," the hypernym might be "**vehicle**." Here, "vehicle" is a more general term that includes "car," "truck," "bike," etc.


Hypernym
Color
Hyponyms
Purple
Red
Blue
Green
Co-Hyponyms
Crimson
Violet
Lavender

Relationships in WordNet

- In WordNet, which is a prominent lexical database for English, synsets and hypernyms are organized in a network. Here's how they **relate**:

Synsets: Words are grouped into synsets based on their synonymous relationships. Each synset provides a definition and usage examples for the words it contains.

Hypernyms: Synsets are linked to their hypernyms, forming a **hierarchical** structure. This helps in understanding the broader context of a word.

For example, the synset for "**car**" might have a hypernym link to "**vehicle**," which in turn has hypernym links to "**transport**," and so on.

C.2 WordNet: A Database of Lexical Relations

WordNet

The most commonly used resource for English sense relations is the **WordNet** lexical database (Fellbaum, 1998). WordNet consists of three separate databases, one each for nouns and verbs and a third for adjectives and adverbs; closed class words are not included. Each database contains a set of lemmas, each one annotated with a set of senses. The WordNet 3.0 release has 117,798 nouns, 11,529 verbs, 22,479 adjectives, and 4,481 adverbs. The average noun has 1.23 senses, and the average verb has 2.16 senses. WordNet can be accessed on the Web or downloaded and accessed locally. Figure C.1 shows the lemma entry for the noun and adjective *bass*.

The noun "bass" has 8 senses in WordNet.
1. bass[1] - (the lowest part of the musical range)
2. bass[2], bass part[1] - (the lowest part in polyphonic music)
3. bass[3], basso[1] - (an adult male singer with the lowest voice)
4. sea bass[1], bass[4] - (the lean flesh of a saltwater fish of the family Serranidae)
5. freshwater bass[1], bass[5] - (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
6. bass[6], bass voice[1], basso[2] - (the lowest adult male singing voice)
7. bass[7] - (the member with the lowest range of a family of musical instruments)
8. bass[8] - (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

The adjective "bass" has 1 sense in WordNet.
1. bass[1], deep[6] - (having or denoting a low vocal or instrumental range)
"a deep voice"; "a bass voice is lower than a baritone voice"; "a bass clarinet"

Figure C.1 A portion of the WordNet 3.0 entry for the noun *bass*.

Practical Applications

- **Word Sense Disambiguation**: By understanding the synsets and hypernyms, NLP systems can better determine which sense of a word is being used in a given context.

- **Semantic Search**: Hypernyms can be used to broaden search queries by including related terms and concepts.

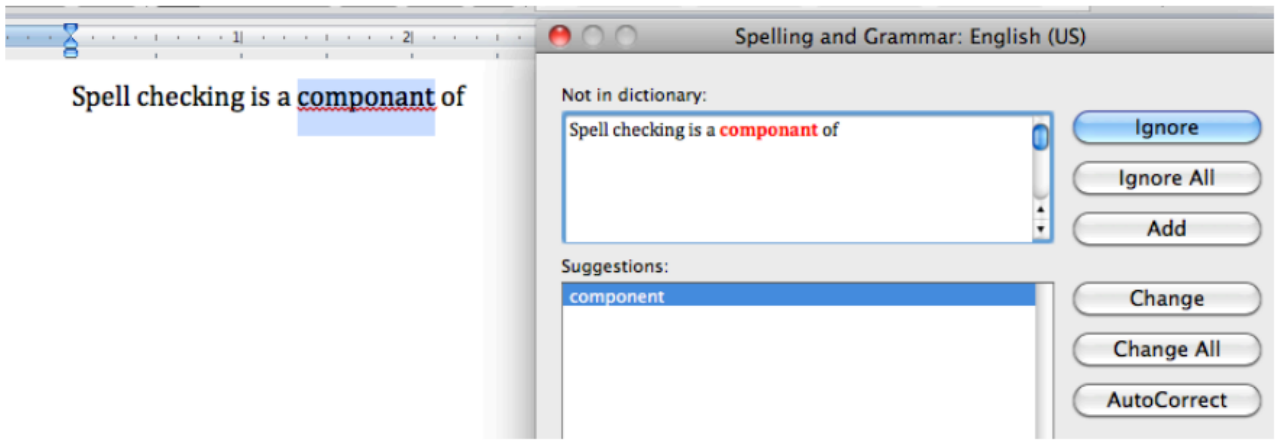
- **Text Classification**: Synsets help in categorizing text based on the concepts conveyed, which can improve the accuracy of text classification models.

The Spelling Correction Task

Dan Jurafsky

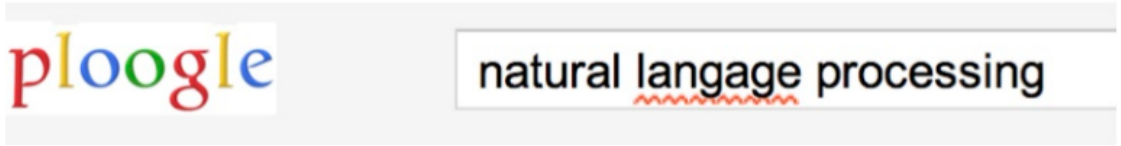
Applications for spelling correction

Word processing



Phones

Web search



Showing results for natural ***language*** processing

Dan Jurafsky

Spelling Tasks

- Spelling Error Detection
- Spelling Error Correction:
  - Autocorrect
    - hte→the
  - Suggest a correction
  - Suggestion lists

Types of spelling errors

- Non-word Errors
  - *graffe* →*giraffe*
- Real-word Errors
  - Typographical errors
    - *three* →*there*
  - Cognitive Errors (homophones)
    - *piece*→*peace*,
    - *too* → *two*

Dan Jurafsky

Rates of spelling errors

26%: Web queries Wang *et al.* 2003

13%: Retyping, no backspace: Whitelaw *et al.* English&German

7%: Words corrected retyping on phone-sized organizer

2%: Words uncorrected on organizer Soukoreff &MacKenzie 2003

1-2%: Retyping: Kane and Wobbrock 2007, Gruden et al. 1983

Dan Jurafsky

Non-word spelling errors

- Non-word spelling error detection:
  - Any word not in a ***dictionary*** is an error
  - The larger the dictionary the better
- Non-word spelling error correction:
  - Generate ***candidates***: real words that are similar to error
  - Choose the one which is best:
    - Shortest weighted edit distance
    - Highest noisy channel probability

Dan Jurafsky

Real word spelling errors

- For each word *w*, generate candidate set:
  - Find candidate words with similar ***pronunciations***
  - Find candidate words with similar ***spelling***
  - Include *w* in candidate set
- Choose best candidate
  - Noisy Channel
  - Classifier

Candidate generation

- Words with similar spelling
  - Small edit distance to error
- Words with similar pronunciation
  - Small edit distance of pronunciation to error

Dan Jurafsky

Damerau-Levenshtein edit distance

- Minimal edit distance between two strings, where edits are:
  - Insertion
  - Deletion
  - Substitution
  - Transposition of two adjacent letters

Words within 1 of acress

Error	Candidate Correction	Correct Letter	Error Letter	Type
acress	actress	t	-	deletion
acress	cress	-	a	insertion
acress	caress	ca	ac	transposition
acress	access	c	r	substitution
acress	across	o	e	substitution
acress	acres	-	s	insertion
acress	acres	-	s	insertion

Candidate generation

- 80% of errors are within edit distance 1
- Almost all errors within edit distance 2

- Also allow insertion of **space** or **hyphen**
  - `thisidea` → `this idea`
  - `inlaw` → `in-law`

HCI issues in spelling

- If very confident in correction
  - Autocorrect
- Less confident
  - Give the best correction
- Less confident
  - Give a correction list
- Unconfident
  - Just flag as an error

Dan Jurafsky

Evaluation

- Some spelling error test sets
  - Wikipedia's list of common English misspelling
  - Aspell filtered version of that list
  - Birkbeck spelling error corpus
  - Peter Norvig's list of errors (includes Wikipedia and Birkbeck, for training or testing)

References

- Slides are heavily inspired from Stanford NLP