Semantic Parsing

1. Introduction

- Two approaches have emerged in the NLP for language understanding.
- In the first approach, a specific, rich meaning representation is created for a limited domain for use by application that are restricted to that domain, such as travel reservations, football game simulations, or querying a geographic database.
- In the second approach, a related set of intermediate-specific meaning representation is created, going from low-level analysis to a middle analysis, and the bigger understanding task is divided into multiple, smaller pieces that are more manageable, such as word sense disambiguation followed by predicate-argument structure recognition.
- Here two types of meaning representations: a domain-dependent, deeper representation and a set of relatively shallow but general-purpose, low-level, and intermediate representation.
- The task of producing the output of the first type is often called deep semantic parsing, and the task of producing the output of the second type is often called shallow semantic parsing.
- The first approach is so specific that porting to every new domain can require anywhere from a few modifications to almost reworking the solution from scratch.
- In other words, the reusability of the representation across domains is very limited.
- The problem with second approach is that it is extremely difficult to construct a general-purpose ontology and create symbols that are shallow enough to be learnable but detailed enough to be useful for all possible applications.
- Ontology means
  1. The branch of metaphysics dealing with the nature of being.
  2. a set of concepts and categories in a subject area or domain that shows their properties and the relations between them.
     "what's new about our ontology is that it is created automatically from large datasets"
- Therefore, an application specific translation layer between the more general representation and the more specific representation becomes necessary.

2. Semantic Interpretation

Semantic parsing can be considered as part of Semantic interpretation, which involves various components that together define a representation of text that can be fed into a computer to allow further computations manipulations and search, which are prerequisite for any language understanding system or application. Here we discuss the structure of semantic theory.

A Semantic theory should be able to:

- Explain sentence having ambiguous meaning: The bill is large is ambiguous in the sense that is could represent money or the beak of a bird.

- Resolve the ambiguities of words in context. The bill is large but need not be paid, the theory should be able to disambiguate the monetary meaning of bill.
- Identify meaningless but syntactically well-formed sentence: Colorless green ideas sleep furiously.
- Identify syntactically or transformationally unrelated paraphrasers of concept having the same semantic content.

- Here we look at some requirements for achieving a semantic representation.

2.1 Structural Ambiguity

- Structure means syntactic structure of sentences.
- The syntactic structure means transforming a sentence into its underlying syntactic representation and in theory of semantic interpretation refer to underlying syntactic representation.

2.2 Word Sense

- In any given language, the same word type is used in different contexts and with different morphological variants to represent different entities or concepts in the world.
- For example, we use the word **nail** to represent a part of the human anatomy and also to represent the generally metallic object used to secure other objects.

intended by the author or speaker. Let's take the following four examples. The presence of words such as *hammer* and *hardware store* in sentences 1 and 2, and of *clipped* and *manicure* in sentences 3 and 4, enable humans to easily disambiguate the sense in which *nail* is used:

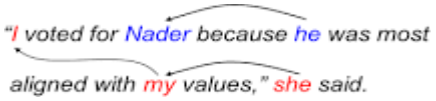
1. He *nailed* the loose arm of the chair with a hammer.
2. He bought a box of *nails* from the hardware store.
3. He went to the beauty salon to get his *nails* clipped.
4. He went to get a manicure. His *nails* had grown very long.

Resolving the sense of words in a discourse, therefore, constitutes one of the steps in the process of semantic interpretation. We discuss it in greater depth in Section 4.4.

2.3 Entity and Event Resolution

- Any discourse consists of a set of entities participating in a series of explicit or implicit events over a period of time.
- So, the next important component of semantic interpretation is the identification of various entities that are sparkled across the discourse using the same or different phrases.

- The predominant tasks have become popular over the years: **named entity recognition** and **coreference resolution**.
- Coreference resolution is the task of finding all expressions that refer to the same entity in a text.



2.4 Predicate Argument Structure

- Once we have the word-sense, entities and events identified, another level of semantics structure comes into play: identifying the participants of the entities in these events.
- Resolving the argument structure of predicate in the sentence is where we identify which entities play what part in which event.
- A word which functions as the verb is called a **predicate** and words which function as the nouns arecalled **arguments**. Here are some other predicates and arguments:

Selena	slept		
argument	predicate		
Tom	is tall		
argument	predicate		
Percy	placed	the penguin on the podium	
argument	predicate	argument	argument

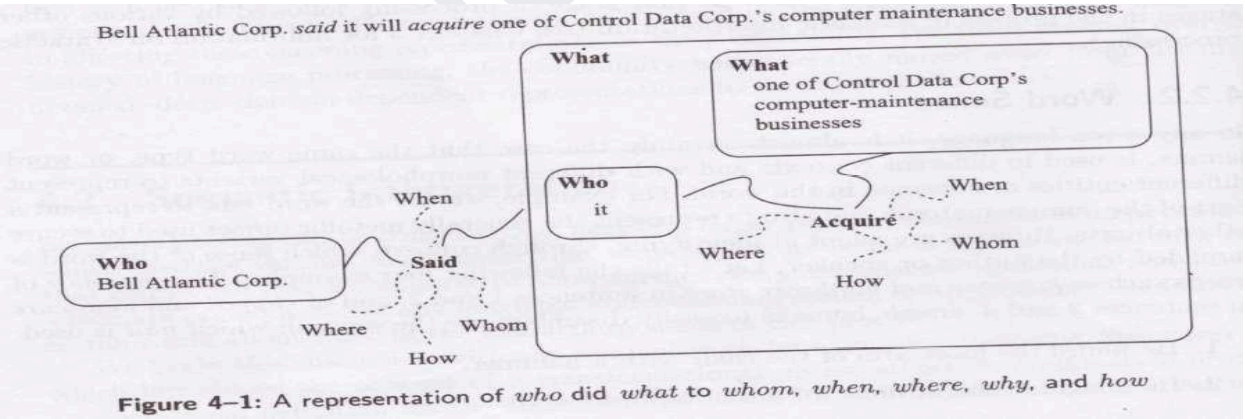


Figure 4–1: A representation of *who* did *what* to *whom*, *when*, *where*, *why*, and *how*

2.5 Meaning Representation

- The final process of the semantic interpretation is to build a semantic representation or meaning representation that can then be manipulated by algorithms to various application ends.
- This process is sometimes called the deep representation.

The following two examples

(1) If our player 2 has the ball, then position our player 5 in the midfield.
`((bowner (player our 2)) (do (player our 5) (pos (midfield)))`

(2) Which river is the longest?
`answer(`x_1`, longest(`x_1 `river(`x_1`)))`

3. System Paradigms

- It is important to get a perspective on the various primary dimensions on which the problem of semantic interpretation has been tackled.
- The approaches generally fall into the following three categories: 1.System architecture

2.Scope 3. Coverage**.**

1. System Architectures

a. **Knowledge based**: These systems use a predefined set of rules or a knowledge base to obtain a solution to a new problem.
b. **Unsupervised**: Thesesystemstendto requireminimal humanintervention to be functional by using existing resources that can be bootstrapped for a particular application or problem domain.
c. **Supervised**: these systems involve the manual annotation of some phenomena that appear in a sufficient quantity of data so that machine learning algorithms can be applied.
d. **Semi-Supervised**: manual annotation is usually very expensive and does not yield enough data to completely capture a phenomenon. In such instances, researches can automatically expand the data set on which their models are trained either by employing machine-generated output directly or by bootstrapping off an existing model by having humans correct its output.

2. Scope:

- **Domain Dependent**: These systems are specific to certain domains, such as air travel reservations or simulated football coaching.
- **Domain Independent**: These systems are general enough that the techniques can be applicable to multiple domains without little or no change.

3. Coverage

a. **Shallow**: These systems tend to produce an intermediate representation that can then be converted to one that a machine can base its action on.
b. **Deep**: These systems usually create a terminal representation that is directly consumed by a machine or application.

4. Word Sense

- **Word Sense** Disambiguation is an important method of **NLP** by which the meaning of a word is determined, which is used in a particular context.
- In a compositional approach to semantics, where the meaning of the whole is

composed on the meaning of parts, the smallest parts under consideration in textual discourse are typically the words themselves: either tokens as they appear in the text or their lemmatized forms.

- Words sense has been examined and studied for a very long time.
- Attempts to solve this problem range from rule based and knowledge based to completely unsupervised, supervised, and semi-supervised learning methods.
- Very early systems were predominantly **rule based or knowledge based** and used dictionary definitions of senses of words.
- **Unsupervised** word sense induction or disambiguation techniques try to induce the senses or word as it appears in various corpora.
- These systems perform either a hard or soft clustering of words and tend to allow the tuning of these clusters to suit a particular application.
- Most recent **supervised** approaches to word sense disambiguation, usually application- independent-level of granularity (including small details). Although the output of supervised approaches can still be amendable to generating a ranking,

  or distribution, of membership sense.
- Word sense ambiguities can be of three principal types: i.**homonymy** ii.**polysemy** iii.**categorial ambiguity**.
- **Homonymy** defined as the words having same spelling or same form but having different and unrelated meaning. For example, the word "Bat" is a homonymy word because bat can be an implement to hit a ball or bat is a nocturnal flying mammal also
- **Polysemy** is a Greek word, which means "many signs". polysemy has the same spelling but different and related meaning.
- Both polysemy and homonymy words have the same syntax or spelling. The main difference between them is that in polysemy, the meanings of the words are related but in homonymy, the meanings of the words are not related.
- For example: Bank Homonymy: financial bank and river bank

  Polysemy: financial bank, bank of clouds and book bank: indicate collection of things.
- **Categorial ambiguity**: the word book can mean a book which contain the chapters or police register which is used to enter the charges against someone.
- In the above note book, text book belongs to the grammatical category of noun and book is verb.
- Distinguishing between these two categories effectively helps disambiguate these two senses.
- Therefore, categorical ambiguity can be resolved with syntactic information (part

of speech) alone, but polyseme and homonymy need more than syntax.

- Traditionally, in English, word senses have been annotated for each part of speech separately, whereas in Chinese, the sense annotation has been done per lemma.

Resources:

- As with any language understanding task, the availability of resources is key factor in the disambiguation of the word senses in corpora.
- Early work on word sense disambiguation used machine readable dictionaries or thesaurus as knowledge sources.
- Two prominent sources were the Longman dictionary of contemporary English (LDOCE) and Roget's Thesaurus.
- The biggest sense annotation corpus OntoNotes released through Linguistic Data Consortium (LDC).
- The Chinese annotation corpus is HowNet.

Systems:

Researchers have explored various system architectures to address the sense disambiguation problem.

We can classify these systems into four main categories: i. rules based or knowledge ii. Supervised iii.unsupervised iv. Semisupervised

Rule Based:

- The first-generation of word sense disambiguation systems was primarily based on dictionary sense definitions.
- Much of this information is historical and cannot readily be translated and made available for building systems today. But some of techniques and algorithms are still available.

- The simplest and oldest dictionary-based sense disambiguation algorithm was introduced by **Lesk.**

The core of the algorithm is that the dictionary senses whose terms most closely overlap with the terms in the context.

Algorithm 4–1 Pseudocode of the simplified Lesk algorithm
The function COMPUTEOVERLAP returns the number of words common to the two sets
Procedure: SIMPLIFIED_LESK(word, sentence) **returns** best sense of *word*

```
1:  best-sense ← most frequent sense of word
2:  max-overlap ← 0
3:  context ← set of words in sentence
4:  for all sense ∈ senses of word do
5:      signature ← set of words in gloss and examples of sense
6:      overlap ← COMPUTEOVERLAP(signature, context)
7:      if overlap gt max-overlap then
8:          max-overlap ← overlap
9:          best-sense ← sense
10:     end if
11: end for
12: return best-sense
```

The Simplified Lesk algorithm

- Let's disambiguate "**bank**" in this sentence:
  The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.
- given the following two WordNet senses:

bank[1]	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	"he cashed a check at the bank", "that bank holds the mortgage on my home"
bank[2]	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

Choose sense with most word overlap between gloss and context
(not counting function words)

The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

bank[1]	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	"he cashed a check at the bank", "that bank holds the mortgage on my home"
bank[2]	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

Another dictionary-based algorithm was suggested Yarowsky.
This study used Roget's Thesaurus categories and classified unseen words into one of these 1042 categories based on a statistical analysis of 100 word concordances for each member of each category.

The method consists of three steps, as shown in Fig below.

- The first step is a collection of contexts.
- The second step computes weights for each of the salient words.

- P(w|Rcat) is the probability of a word w occurring in the context of a Roget's Thesaurus category Rcat.
- P(w|Rcat) |Pr(w) , the probability of a word (w) appearing in the context of a Roget category divided by its overall probability in the corpus.
- Finally, in third step, the unseen words in the test set are classified into the category that has the maximum weight.

1. Collect contexts for each of the *Roget's Thesaurus* categories.
2. Determine weights for each of the salient words in the context.

$$\frac{P(w_i|RCat)}{P(w_i)}$$

3. Use the weights for predicting the appropriate category of the word in the test corpus.

$$\arg\max_{RCat} \sum_{W} \log \frac{P(w_i|RCat)P(RCat)}{P(w_i)}$$

Figure 4–2: Algorithm for disambiguating words into *Roget's Thesaurus* categories

WALKER'S ALGORITHM

- A Thesaurus Based approach.
- ***Step 1:*** *For each sense of the target word find the thesaurus category to which that sense belongs.*
- ***Step 2:*** *Calculate the score for each sense by using the context words. A context words will add 1 to the score of the sense if the thesaurus category of the word matches that of the sense.*
  - *E.g. The money in this **bank** fetches an interest of 8% per annum*
  - Target word: ***bank***
  - Clue words from the context: ***money, interest, annum, fetch***

	Sense1: Finance	Sense2: Location
Money	+1	0
Interest	+1	0
Fetch	0	0
Annum	+1	0
Total	3	0

Context words add 1 to the sense when the topic of the word matches that of the sense

Supervised:

- The simpler form of word sense disambiguating systems the supervised approach, which tends to transfer all the complexity to the machine learning machinery while still requiring hand annotation tends to be superior to unsupervised and performs best when tested on annotated data.
- These systems typically consist of a machine learning classifier trained on various features extracted for words that have been manually disambiguated in a given corpus and the application of the resulting models to disambiguating words in the

unseen test sets.

- A good feature of these systems is that the user can incorporate rules and knowledge in the form of features.

Classifier:

Probably the most common and high performing classifiers are support vector machine (SVMs) and maximum entropy classifiers.

Features: Here we discuss a more commonly found subset of features that have been useful in supervised learning of word sense.

Lexical context: The feature comprises the words and lemma of words occurring in the entire paragraph or a smaller window of usually five words.

Parts of speech: the feature comprises the surrounding the word that is being sense tagged. **Bag of words context**: this feature comprises using an unordered set of words in the context window.

Local Collocations: Local **collocations** are an ordered sequence of phrases near the target word that provide semantic context for disambiguation. Usually, a very small window of about three tokens on each side of the target word, most often in contiguous pairs or triplets, are added as a list of features.

Syntactic relations: if the parse of the sentence containing the target word is available, then we can use syntactic features.

Topic features: The board topic, or domain, of the article that word belongs to is also a good indicator of what sense of the word might be most frequent.

is also a good indicator of what sense ...

Chen and Palmer [51] recently proposed some additional rich features for disambiguation:

Voice of the sentence—This ternary feature indicates whether the sentence in which the word occurs is a passive, semipassive,[3] or active sentence.

Presence of subject/object—This binary feature indicates whether the target word has a subject or object. Given a large amount of training data, we could also use the actual lexeme and possibly the semantic roles rather than the syntactic subject/objects.

Sentential complement—This binary feature indicates whether the word has a sentential complement.

Prepositional phrase adjunct—This feature indicates whether the target word has a prepositional phrase, and if so, selects the head of the noun phrase inside the prepositional phrase.

Named entity—This feature is the named entity of the proper nouns and certain types of common nouns.

WordNet—WordNet synsets of the hypernyms of head nouns of the noun phrase arguments of verbs and prepositions.

following research in semantic role labeling. Dligach and Palmer [52]

"Word sense disambiguation" (WSD) is a natural language processing (NLP) task that involves determining the correct sense or meaning of a word within a given context. Many words in natural language have multiple meanings or senses, and WSD aims to choose the most appropriate sense for a word in a specific sentence or context.

Supervised learning with Support Vector Machines (SVM) is one approach to solving the WSD problem. Here's how it works:

- **Data Collection**: To train an SVM for WSD, you need a labeled dataset where each word is tagged with its correct sense in various contexts. This dataset is typically created by human annotators who assign senses to words in sentences.
- **Feature Extraction**: For each word in the dataset, you need to extract relevant features from its context. These features could include the words surrounding the target word, part-of-speech tags, syntactic information, and more. These features serve as the input to the SVM.
- **Training**: Once you have the labeled dataset and extracted features, you can train an SVM classifier. The goal is to teach the SVM to learn patterns in the features that are indicative of specific word senses.
- **Testing/Predicting**: After training, you can use the SVM to predict the sense of an ambiguous word in a new, unseen sentence. The SVM considers the context features and assigns the word the most likely sense based on what it learned during training.
- **Evaluation**: To assess the performance of your WSD system, you can use various evaluation metrics, such as accuracy, precision, recall, and F1-score. These metrics help you measure how well your SVM-based WSD system is performing in disambiguating word senses.

SVMs are popular for WSD because they are effective at handling high-dimensional feature spaces and can learn complex decision boundaries. However, the success of the SVM-based WSD system heavily depends on the quality of the labeled dataset and the choice of features used for training.

Algorithm 4–2 Rules for selecting syntactic relations as features

```
if w is a noun then
    select parent head word (h)
    select part of speech of h
    select voice of h
    select position of h (left, right)
else if w is a verb then
    select nearest word l to the left of w such that w is the parent head word of l
    select nearest word r to the right of w such that w is the parent head word of r
    select part of speech of l
    select part of speech of r
    select part of speech of w
    select voice of w
else if w is a adjective then
    select parent head word (h)
    select part of speech of h
end if
```

The identification of the head word is important in syntax because it helps determine the grammatical structure of a phrase or sentence. For feature selection in NLP tasks like parsing or word sense disambiguation, knowing the head word and its relationships with other words in a

sentence can be valuable information. Syntactic relations often involve the relationship between a head word and its dependents or modifiers, and these relations can be used as features in various natural language processing applications.

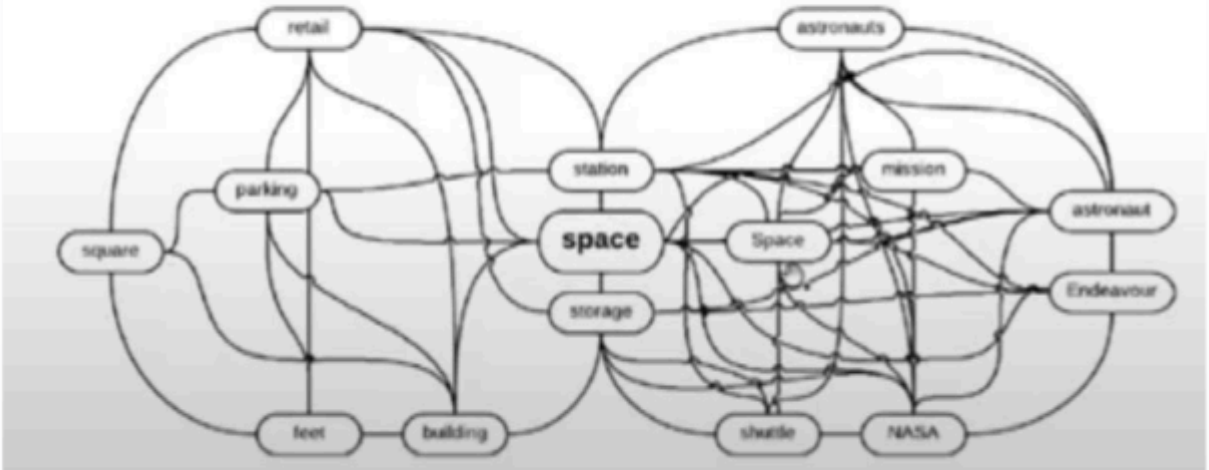
Unsupervised:

Unsupervised learning in Natural Language Processing (NLP) is a category of machine learning where the model is trained on unlabeled data without explicit supervision or predefined categories. It aims to discover patterns, structures, or representations within the data. One concept related to unsupervised learning in NLP is "Conceptual Density."

HyperLex

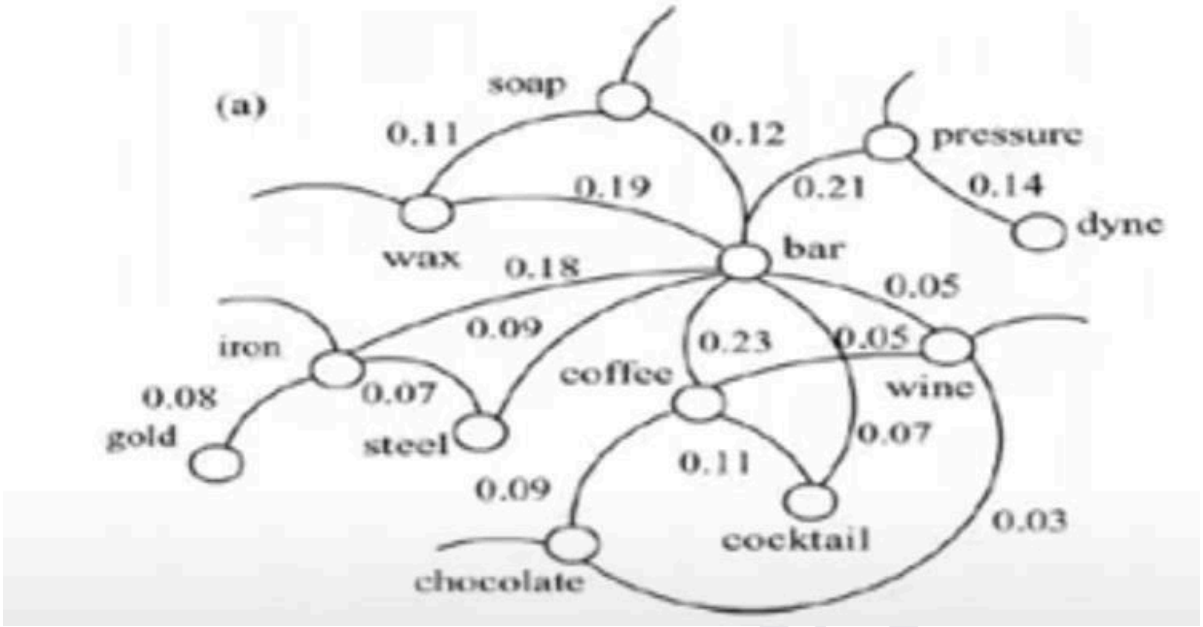
Key Idea: Word Sense Induction

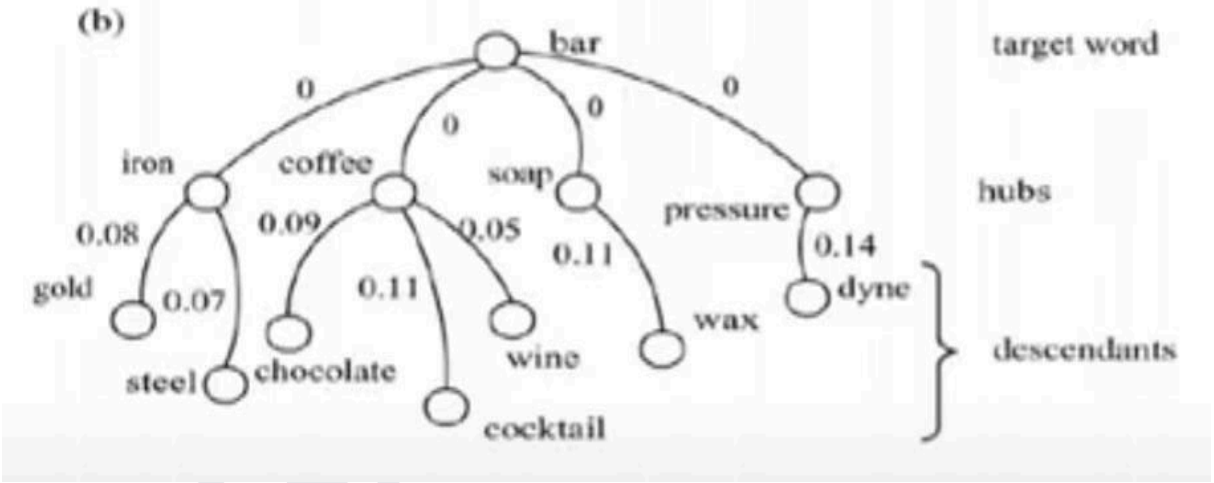
- Instead of using "dictionary defined senses", extract the "senses from the corpus" itself
- These "corpus senses" or "uses" correspond to clusters of similar contexts for a word.



Detecting Root Hubs

- Different uses of a target word form highly interconnected bundles (or high density components)
- In each high density component one of the nodes (hub) has a higher degree than the others.
- **Step 1:** Construct co-occurrence graph, G.
- **Step 2:** Arrange nodes in G in decreasing order of degree.
- **Step 3:** Select the node from G which has the highest degree. This node will be the hub of the first high density component.
- **Step 4:** Delete this hub and all its neighbors from G.
- **Step 5:** Repeat Step 3 and 4 to detect the hubs of other high density components



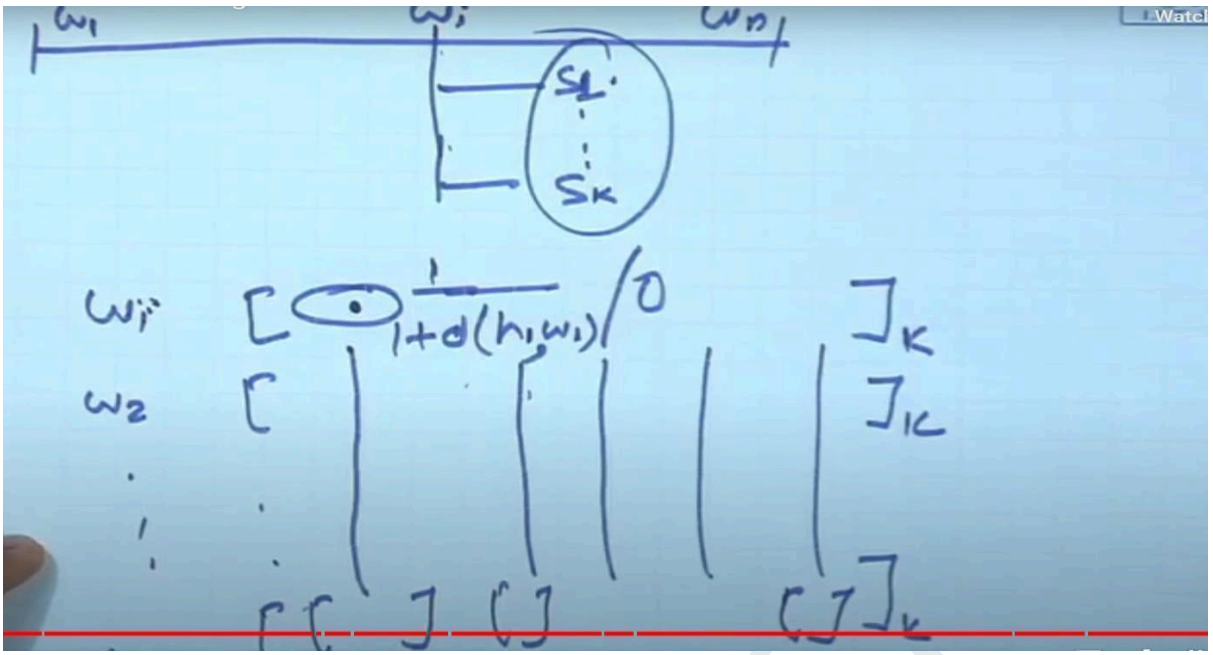


- Attach each node to the root hub closest to it.
- The distance between two nodes is measured as the smallest sum of weights of the edges on the paths linking them.

Computing distance between two nodes w_i and w_j

$$w_{ij} = 1 - max\{P(w_i|w_j), P(w_j|w_i)\}$$

where $P(w_i|w_j) = \frac{freq_{ij}}{freq_j}$



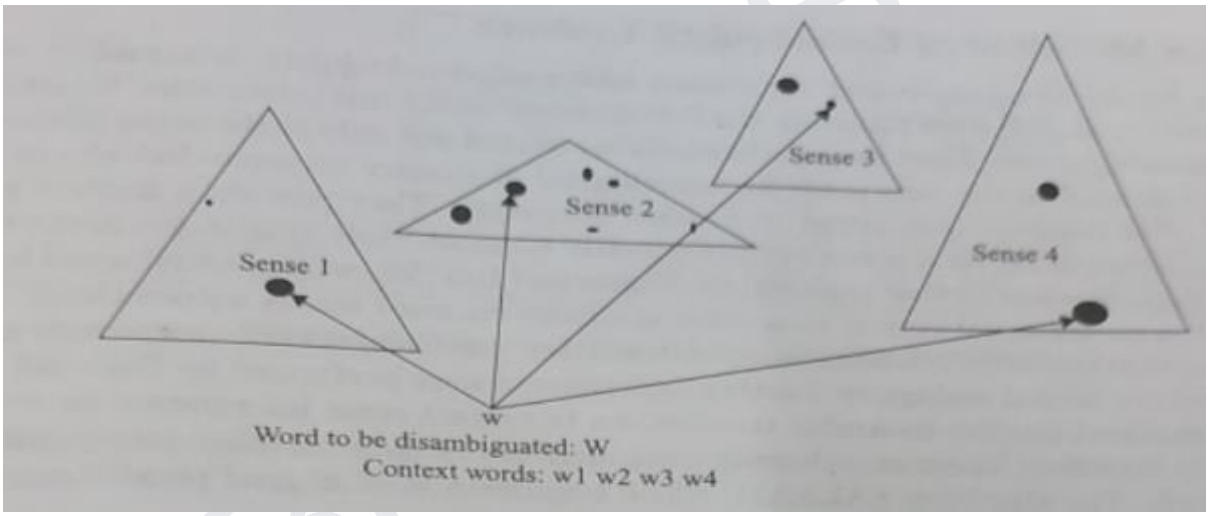


Figure: Conceptual Density

Semi Supervised:

Semi-supervised learning is a machine learning paradigm that combines both labeled and unlabeled data to improve model performance. In the context of word sense disambiguation

(WSD) in Natural Language Processing (NLP), semi-supervised learning techniques can be quite beneficial because labeled data for WSD is often limited and expensive to obtain. Here's an overview of a semi-supervised learning algorithm for WSD:

Self-Training for WSD:

Self-training is a popular semi-supervised learning approach that can be adapted for WSD. In self-training for WSD, you start with a small set of labeled examples and a larger set of unlabeled examples. The process involves iterative steps:

- **Initialization**: Begin with a small labeled dataset where each example consists of a sentence containing an ambiguous word and its corresponding sense label.
- **Feature Extraction**: Extract relevant features from the labeled examples, which typically include information about the target word, its context words, part-of-speech tags, syntactic relations, and more.
- **Model Training**: Train a WSD model using the labeled data. This can be a supervised machine learning model like Support Vector Machines (SVM), Naive Bayes, or a neural network-based model.
- **Prediction**: Use the trained model to predict word senses for the unlabeled data. Apply the model to the sentences containing the ambiguous word from the unlabeled dataset to assign senses to those instances.
- **Confidence Threshold**: Introduce a confidence threshold or some criteria to filter the predictions. For instance, you can choose to keep only the predictions where the model is highly confident.
- **Adding Labeled Data**: Add the confidently predicted examples to the labeled dataset, marking them as newly labeled instances.
- **Iteratio**: Repeat steps 2-6 for a fixed number of iterations or until convergence.
- **final Model**: Train a final model using the combined labeled data (original labeled dataset plus the newly labeled instances) to create a more robust WSD model.

Advantages of Self-Training for WSD:

- It leverages a larger pool of unlabeled data, which can be especially beneficial when labeled data is scarce.
- It allows the model to learn from its own predictions and iteratively improve.
- Self-training is a flexible approach and can be used with various machine learning models.

Challenges:

- Labeling errors: The initial labeled dataset should be of high quality because errors can accumulate during self-training iterations.

Semi-supervised learning with self-training can be effective for WSD, but it's essential to carefully design the process, monitor model performance, and apply filtering criteria to ensure the quality of the added labeled instances.

Motivation and concept of Yorowsky algorithm

- Annotations are expensive!
- "Bootstrapping" or co-training
  - Start with (small) seed, learn decision list
  - Use decision list to label rest of corpus
  - Retain 'confident' labels, treat as annotated data to learn new decision list
  - Repeat ...
- Heuristics (derived from observation):
  - One sense per discourse
  - One sense per collocation

One Sense per Discourse

- A word tends to preserve its meaning across all its occurrences in a given discourse

One Sense per Collocation

- A word tends to preserve its meaning when used in the same collocation
  - Strong for adjacent collocations
  - Weaker as the distance between the words increases

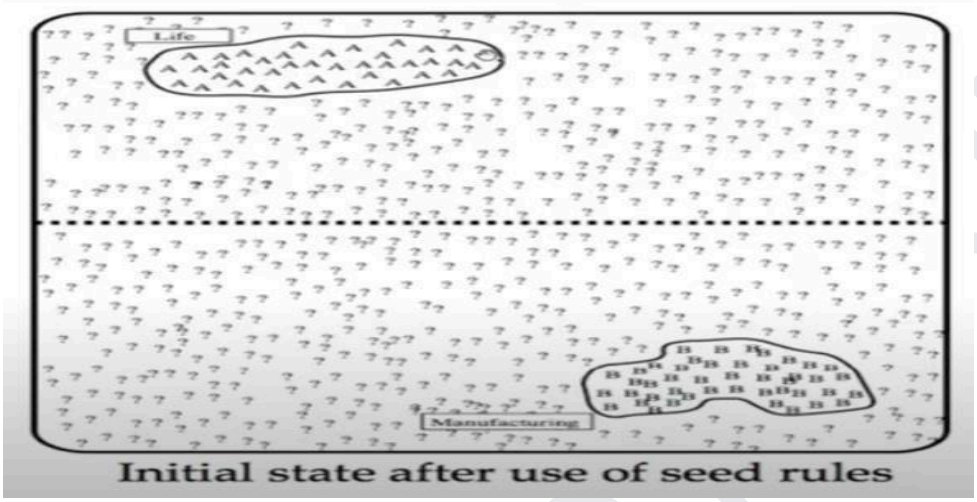
Example

- Disambiguating plant (industrial sense) vs. plant (living thing sense)
- Think of seed features for each sense
  - Industrial sense: co-occurring with 'manufacturing'
  - Living thing sense: co-occurring with 'life'
- Use 'one sense per collocation' to build initial decision list classifier
- Treat results (having high probability) as annotated data, train new decision list classifier, iterate

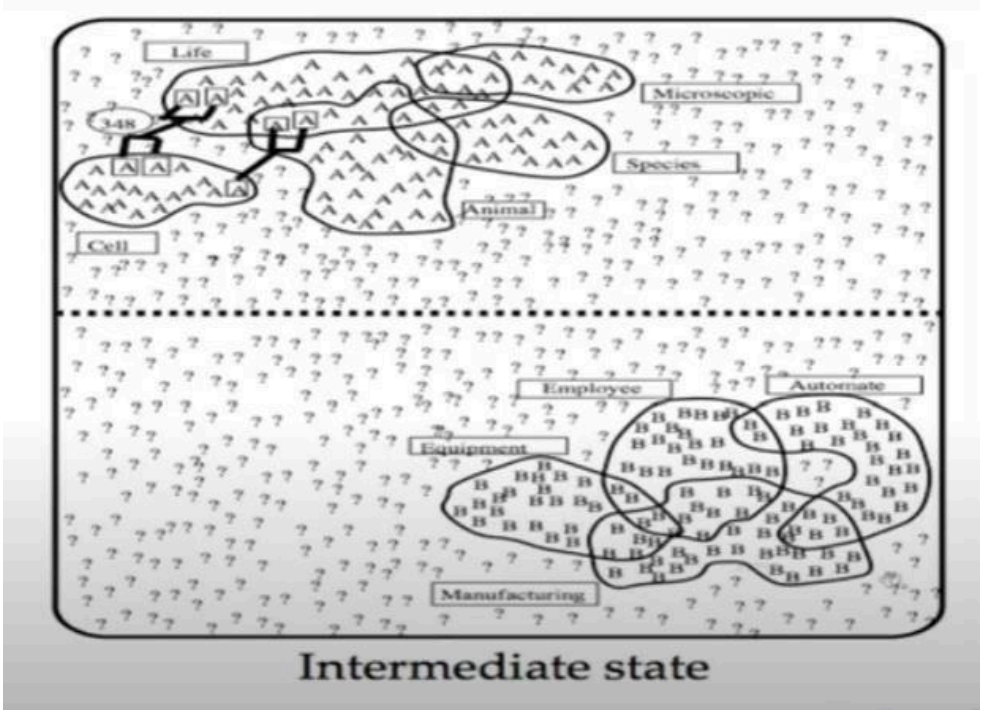
used to strain microscopic plant life from the
zonal distribution of plant life .
close-up studies of plant life and natural
too rapid growth of aquatic plant life in water
the proliferation of plant and animal life
establishment phase of the plant virus life cycle
that divide life into plant and animal kingdom
many dangers to plant and animal life
mammals . Animal and plant life are delicately

automated manufacturing plant in Fremont
vast manufacturing plant and distribution
chemical manufacturing plant , producing viscose
keep a manufacturing plant profitable without
computer manufacturing plant and adjacent
discovered at a St. Louis plant manufacturing
copper manufacturing plant found that they
copper wire manufacturing plant , for example

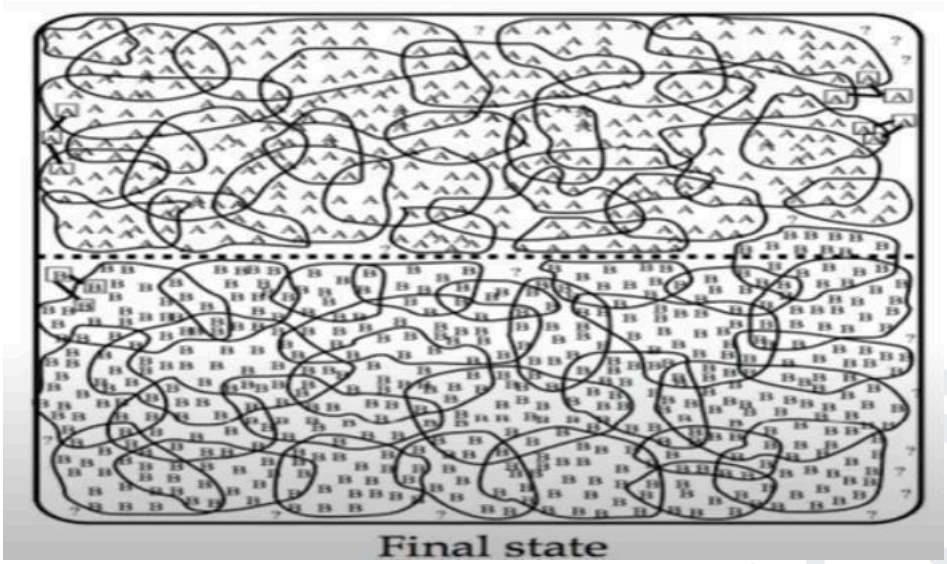
vinyl chloride monomer plant , which is
molecules found in plant and animal tissue
Nissan car and truck plant in Japan is
and Golgi apparatus of plant and animal cells
union responses to plant closures .
cell types found in the plant kingdom are
company said the plant is still operating
Although thousands of plant and animal species
animal rather than plant tissues can be



Initial state after use of seed rules



Intermediate state



Termination

- Stop when
  - Error on training data is less than a threshold
  - No more training data is covered
- Use final decision list for WSD

Advantages

- Accuracy is about as good as a supervised algorithm
- Bootstrapping: far less manual effort

Figure: Yorowsky algorithm

Step 1. In a sufficiently large corpus, identify all the instances of a particular polysemous word that needs to be disambiguated, storing its context alongside.
Step 2. Identify a small set of instances that are strongly representative of one of the senses of the word. This can either be done in a completely unsupervised fashion by identifying collocations that give a strong indication of the sense usage for the word under consideration or by manually tagging a small portion of the data. In this example, we assume a polysemous word with only two senses, but this algorithm can be extended to n senses.
Step 3.

Step 3a. Train a supervised classifier on this set of examples.

Step 3b. Using these classifiers, classify the remaining instances of the word in the corpus and select those that are classified above a certain level of confidence.

Step 3c. Filter out the possible misclassifications using one sense per discourse constraint, and identify possible new collocations to be added to the list of seed collocations.

Step 3d. Repeat step 3 iteratively, thereby slowly shrinking the residual.

Step 4. Stop. At some point, a small, stable residual will remain.
Step 5. The trained classifier can now be used to classify new data, and that in turn can be used to annotate the original corpus with sense tags and probabilities.