Trade-Offs in Various Approaches to NLP

Natural Language Processing

Choosing Your Approach in NLP

Some major trade-offs:

- Shallow vs. deep
- Statistical vs. symbolic
- Feature engineering vs. feature learning
- Top-down vs. bottom-up
- Transparent vs. opaque (AI vs. XAI)

Choosing Your Approach in NLP

The question we are asking is:

- Shallow vs. deep
  - ***Shall I build a very robust, exhaustive representation of text meaning, or shall I just pick out a few elements needed for my application?***
- Statistical vs. symbolic
- Feature engineering vs. feature learning
- Top-down vs. bottom-up
- Transparent vs. opaque (AI vs. XAI)

Choosing Your Approach in NLP

The question we are asking is:

- Shallow vs. deep
- Statistical vs. symbolic
  - ***Shall I leverage powerful, complex statistical methods, or use rules that can be operated on by strict logic?***
- Feature engineering vs. feature learning
- Top-down vs. bottom-up
- Transparent vs. opaque (AI vs. XAI)

Choosing Your Approach in NLP

The question we are asking is:

- Shallow vs. deep
- Statistical vs. symbolic
- Feature engineering vs. feature learning
  - ***Shall I involve human experts to engineer features, or shall I use them only to establish a training set?***
- Top-down vs. bottom-up
- Transparent vs. opaque (AI vs. XAI)

Choosing Your Approach in NLP

The question we are asking is:

- Shallow vs. deep
- Statistical vs. symbolic
- Feature engineering vs. feature learning
- Top-down vs. bottom-up
  - ***Shall I start with high-level classifications and text characteristics, then gradually break them down into more detail, or shall I do the reverse?***
- Transparent vs. opaque (AI vs. XAI)

Choosing Your Approach in NLP

The question we are asking is:

- Shallow vs. deep
- Statistical vs. symbolic
- Feature engineering vs. feature learning
- Top-down vs. bottom-up
- Transparent vs. opaque (AI vs. XAI)
    - ***Shall I engineer a system that is explainable to an intelligent SME, or is it okay to build something only my fellow data scientists and engineers could understand?***

DataScience@SMU

Shallow vs. Deep NLP

Natural Language Processing

Choosing Your Approach in NLP

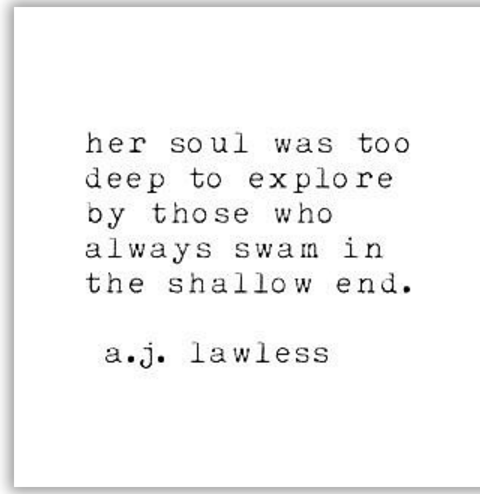
The question we are asking is:

- Shallow vs. deep
    - ***Shall I build a very robust, exhaustive representation of text meaning, or shall I just pick out a few elements needed for my application?***
- Statistical vs. symbolic
- Feature engineering vs. feature learning
- Top-down vs. bottom-up
- Transparent vs. opaque (AI vs. XAI)

Shallow vs. Deep

Across all the branches of AI, the "deep vs. shallow" distinction is very difficult to master because it is applied at very different junctures within AI, and even within NLP particularly.

You may have heard of the distinction in machine learning (ML), where we hear about "deep learning" as opposed to "shallow learning."

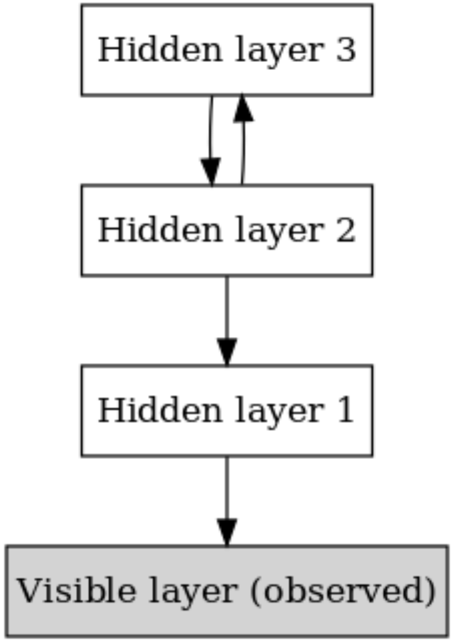


Quotes like this illustrate how the word "deep" usually has a positive connotation, and "shallow" negative—but don't be fooled by this!

Shallow vs. Deep

In ML, “deep learning” refers to systems having a lot of hidden layers.

- There is no standard as to how many layers count as “a lot,” but generally it is at least three.
- These layers often are in neural networks but may be in something else such as a DBN (deep belief network).

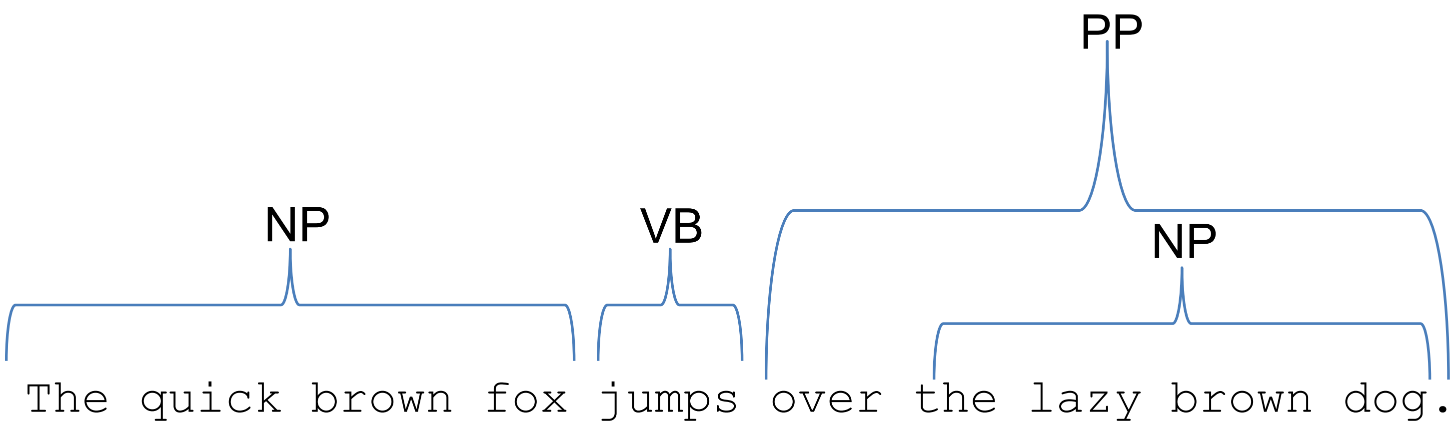


In those modules of NLP where we utilize ML, we might use the terminology in this same way. But we have other ways.

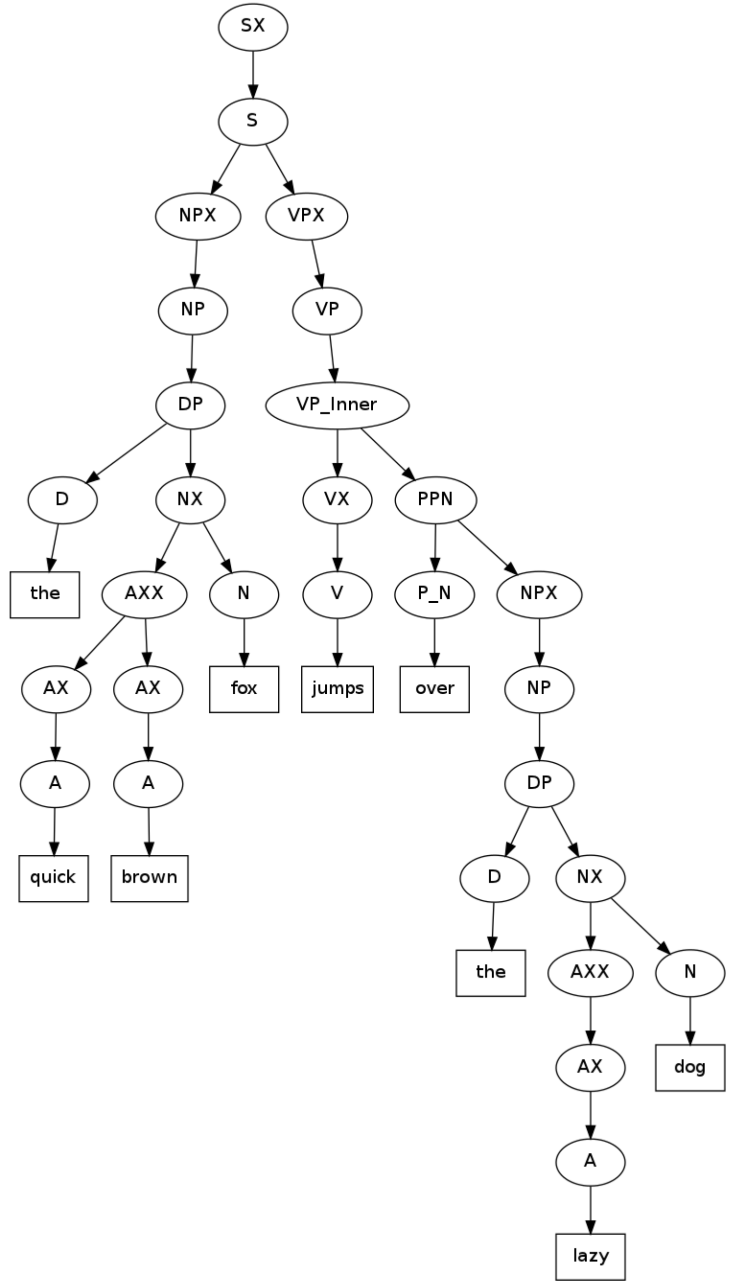
Shallow vs. Deep

On the syntactic level, a parser that stops short of the most granular level of detail is called a "shallow" parser.

- It's also called a "chunk" parser because it usually leaves parts of the input sentence in multiword chunks.
- For example, it may leave multiword noun phrases intact as single units and not break them down into their determiners, adjectives, etc.



Compare to what a full (deep) parse tree would be for the same sentence!



Shallow vs. Deep

On the semantic level, a shallow semantic parser attributes thematic roles to phrases surrounding a target word.

- So our system might label one noun phrase the “agent" and another the "recipient."
- It does not give great specificity as to what these roles are; e.g., the recipient might be catching a ball, receiving a letter in the mail, or be the butt of a joke, and the system does not differentiate these. *And that’s why it’s "shallow."*

This example of shallow semantic parsing is from The Stanford Natural Language Processing Group

For example, the sentence

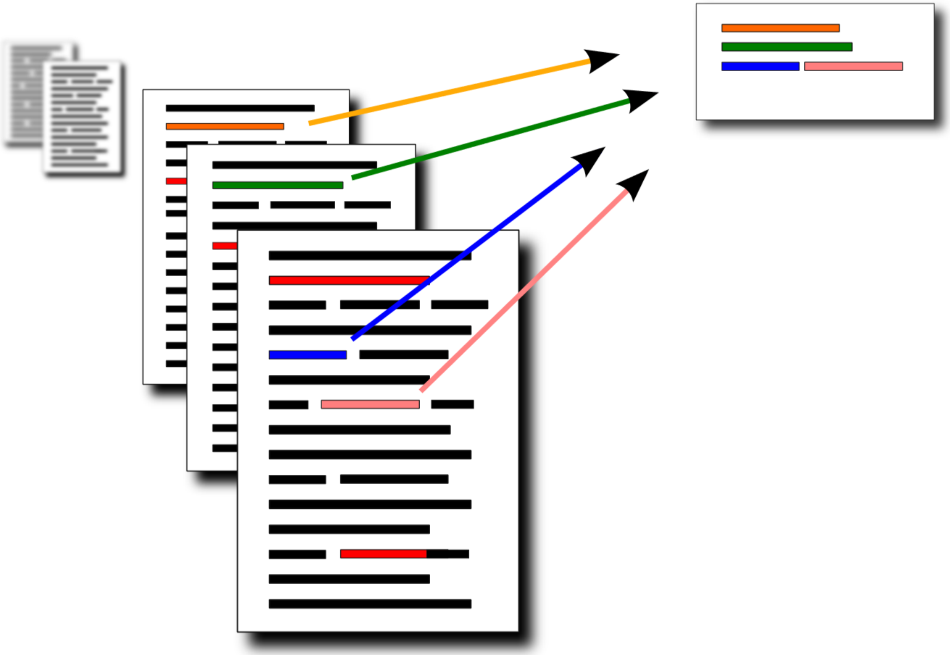
- Shaw Publishing offered Mr. Smith a reimbursement last March.

Is labeled as:

- [$_{\text{AGENT}}$Shaw Publishing] **offered** [$_{\text{RECEPIENT}}$Mr. Smith] [$_{\text{THEME}}$a reimbursement] [$_{\text{TIME}}$last March] .

Shallow vs. Deep

Finally, in implementing many useful types of semantic analysis, we only produce a partial representation of each sentence, paragraph, document, or corpus.

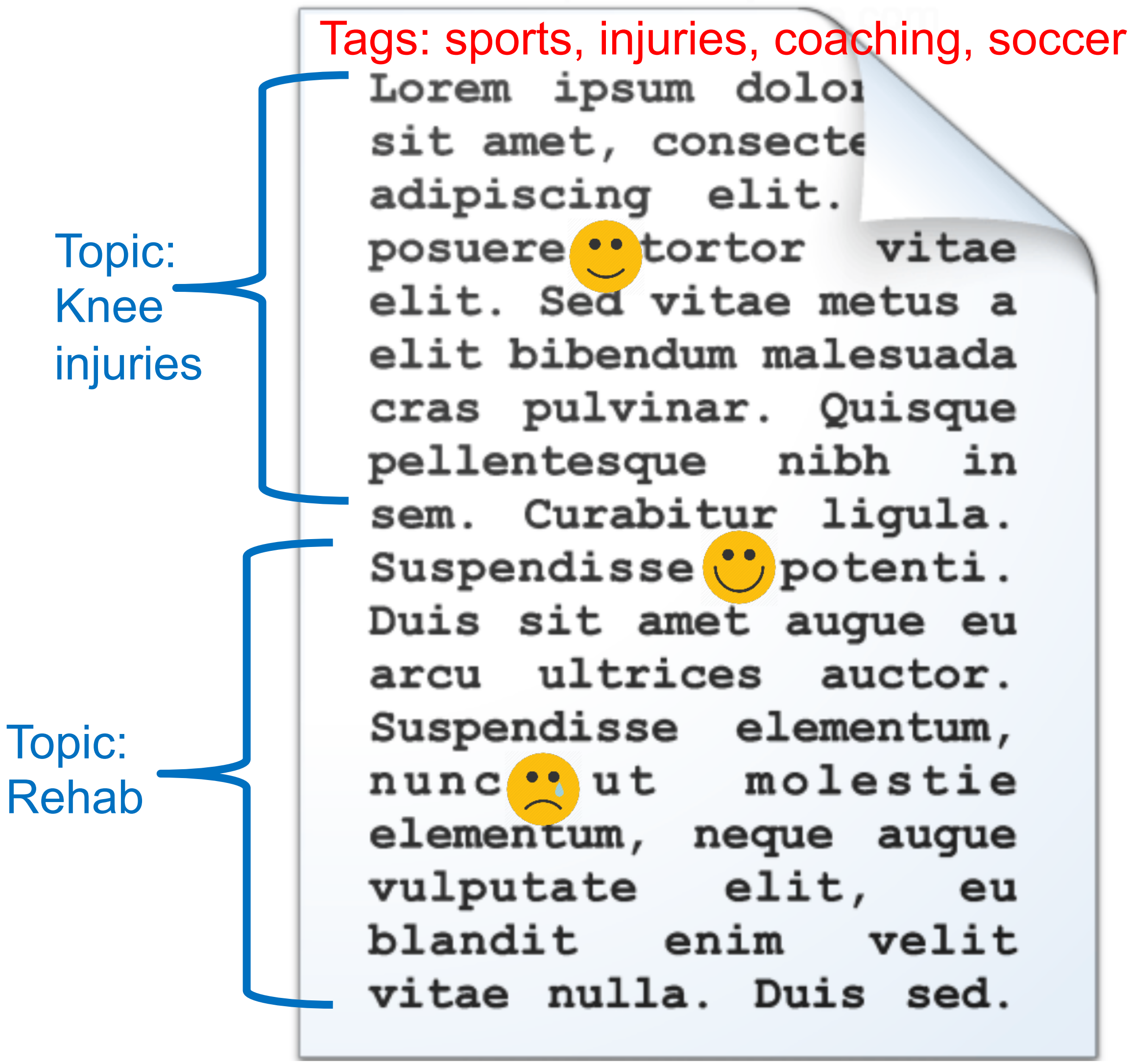


Because the representation is not semantically exhaustive (does not represent all the meanings of all the content—usually not anywhere near it, in fact), it is "shallow" semantics.

Shallow vs. Deep

Remember this example?

Merely performing tagging, topic segmentation, and sentiment analysis, even in combination, is an example of "shallow semantics."



Should I Go Deep or Shallow? Yes!

There are strengths to both!

COMMUNICATIONS OF THE ACM

HOME | CURRENT ISSUE | NEWS | BLOGS | OPINION | RESEARCH | PRA

Home / Magazine Archive / March 2016 (Vol. 59, No. 3) / Deep or Shallow, NLP Is Breaking Out / Abstract

NEWS

Deep or Shallow, NLP Is Breaking Out

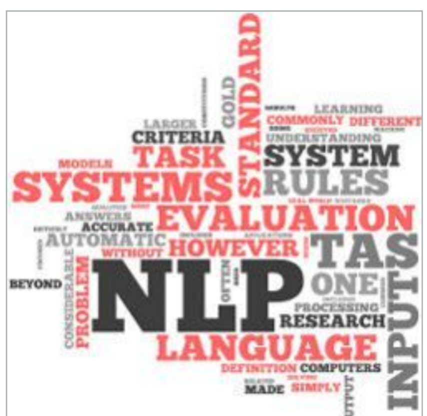
By Gregory Goth

Communications of the ACM, Vol. 59 No. 3, Pages 13-16

10.1145/2874915

Comments (1)

VIEW AS: SHARE:



One of the featured speakers at the inaugural Text By The Bay conference, held in San Francisco in April 2015, drew laughter when describing a neural network question-answering model that could beat human players in a trivia game.

While such performance by computers is fairly well known to the general public, thanks to IBM's Watson cognitive computer, the speaker, natural language processing (NLP) researcher Richard Socher, said, the neural network model he described "was built by one grad student using deep learning" rather than by a large team with the resources of a global corporation behind them.

Shallow vs. Deep

Deep semantics, *in the strictest sense*, is really a "Holy Grail" of NLP in that it implies a machine that can represent everything about a text that a competent human being would.

- We rarely even attempt that.
- So doing “deep” NLP is really a matter of degree, with no clear dividing line.

DataScience@SMU

Statistical vs. Symbolic NLP

Natural Language Processing

Choosing Your Approach in NLP

The question we are asking is:

- Shallow vs. deep
- Statistical vs. symbolic
  - ***Shall I leverage powerful, complex statistical methods, or use rules that can be operated on by strict logic?***
- Feature engineering vs. feature learning
- Top-down vs. bottom-up
- Transparent vs. opaque (AI vs. XAI)

Statistical vs. Symbolic

In symbolic NLP we have a rule-based system whereby discrete tests are performed to apply each rule.

- The tests are based upon *necessary and sufficient conditions* for a result.
- We are using *Boolean logic* and/or particular numerical thresholds to apply criteria to textual input.

```
If (P and Q), then R
P
Q
------------
Therefore, R
```

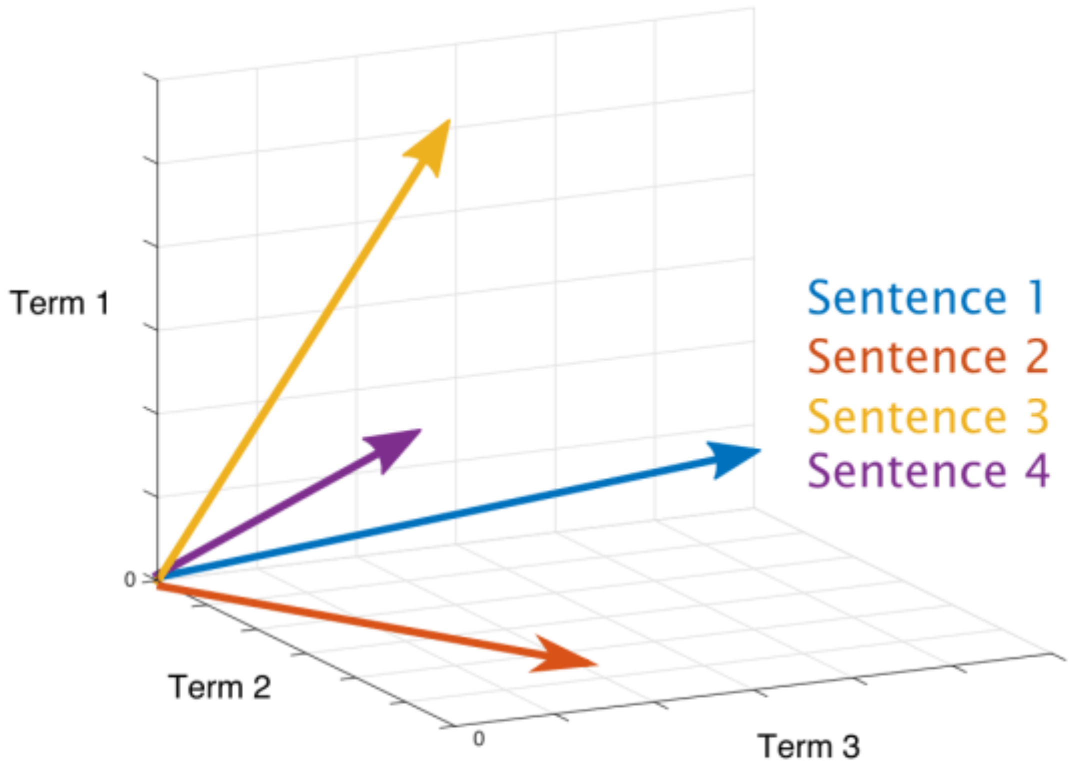
```
If F(x)≥ 0.4
  and G(x)≤ 0.7,
    then R
F(x) = 0.5
G(x) = 0.3
------------
Therefore R
```

Statistical vs. Symbolic

In statistical NLP we use a combination of signals or nodes of varying strengths.

The net result of many combined signals gives us a final statistical score (or profile or vector).

In this example, sentences whose term-occurrence vectors are the closest in feature space are deemed therefore to be the most similar to each other.



Which Is Better?

- Let's consider an application where we want to build a "similar words" index for a particular corpus, let us say of medical documents.
  - We could use co-occurrence, indirect co-occurrence, or similar statistical methods.
  - A high-powered version of this is called "latent semantic analysis" or "latent semantic indexing" (LSA/LSI).

Blah blah doctors and nurses blah blah blah patients seeing nurses blah blah patients seeing doctors blah blah blah blah doctors working at clinics blah blah blah blah blah nurses working at clinics, blah blah physicians' salaries blah blah nurses on strike.

What Happens When We Go Statistical?

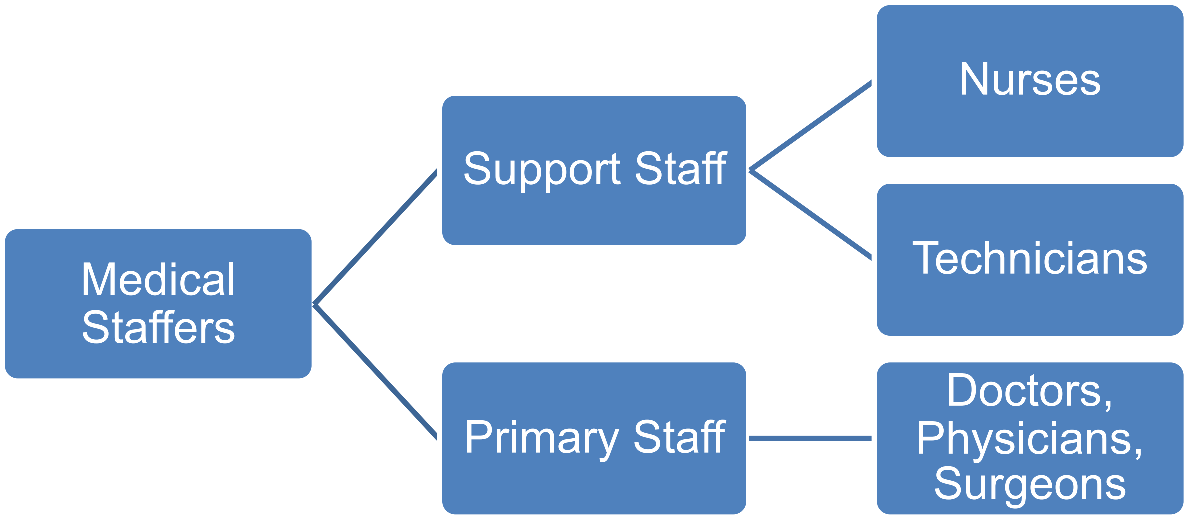
Whereas most of us would agree that “physician” is closer in meaning to the word “doctor” than is the word “nurse,” when LSA is employed, it may happen that “nurse” is found to be more strongly related statistically than is the word “physician.”

- In effect, the statistical method incorrectly suggests that “nurse” is more closely related in meaning to “doctor” than “physician” is.
- Oops!

> Blah blah doctors and nurses blah blah blah *patients seeing nurses* blah blah *patients seeing doctors* blah blah blah blah *doctors working at clinics* blah blah blah blah blah nurses working at clinics, blah blah physicians’ salaries blah blah nurses on strike.

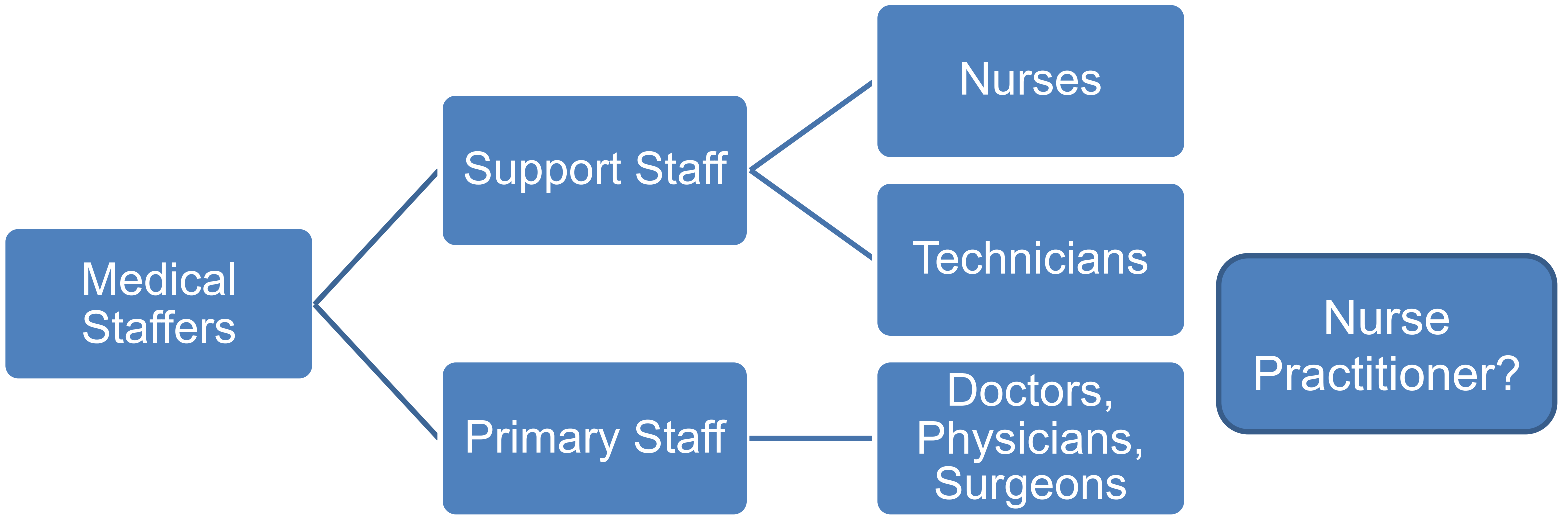
What Happens When We Go Symbolic?

- Another approach, which avoids this error, is to create a hand-crafted taxonomy that separates doctors from nurses.
- This could help us group related terms where they seem to belong, taxonomically speaking. We could use this taxonomy, not LSA, to build an index of related terms.



What Happens When We Go Symbolic?

- But what happens when we encounter a new term called a “nurse practitioner” (one who can prescribe medications and see patients independently)?
  - Should it be a subtype of nurse, or of doctor, or a new category unto itself?
  - Whatever is decided, it is a challenge to manually update the taxonomy in reaction to the ever-expanding world of information.



So Which Is Better?

- Clearly, the rule-based and statistical approaches both have inherent strengths and weaknesses.
- Statistical approaches give us scalability and rapid learning, while rule-based methods give us transparency, control, and an easy way to implement a manual override to an undesired output.

Statistical NLP

Symbolic NLP

Easily scalable

Manually controllable

DataScience@SMU

Feature Engineering vs. Feature Learning in NLP

Natural Language Processing

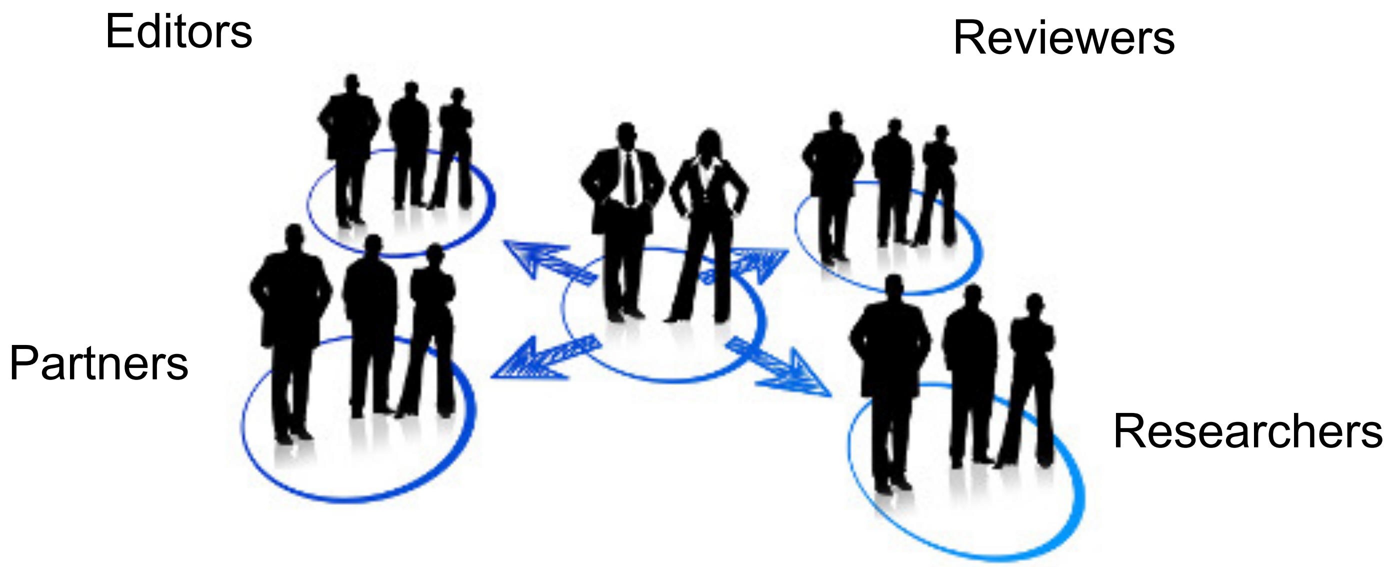
Choosing Your Approach in NLP

The question we are asking is:

- Shallow vs. deep
- Statistical vs. symbolic
- Feature learning vs. feature engineering
  - ***Shall I involve human experts to engineer features, or shall I use them only to establish a training set?***
- Top-down vs. bottom-up
- Transparent vs. opaque (AI vs. XAI)

Who Are Your Stakeholders?

- In many cases you stand in the middle of multiple stakeholders who are subject matter experts (SMEs).
- Sometimes they want a hands-off role in your NLP project—they just want you to deliver a totally automated tool.
- In other cases they want you to give them a measure of editorial control—essentially to engage them in feature engineering.



Feature Learning

- To try to keep the SME's hands off, you start with a collection of already annotated documents (training set), and feed them into a classifier that can learn its own features, such as one based on a Support Vector Machine (SVM).
- Deficiencies in precision or recall are addressed by expanding the training set.

PROS:
- Very convenient when a training set already exists (or can be collected passively)
- Avoids your having to work with SMEs!

CONS:
- If you have to ask SMEs to create a training set, it can be painful.
- Your SMEs don't feel that they "own" how the AI behaves. It's a "black box" to them.

Feature Engineering

- To let the SMEs be hands on, you draft them for feature engineering. They must be highly competent in the domain of documents you are classifying.
- Working with a knowledge engineer, the SMEs brainstorm features from the raw text that might be clues to classification.
- Deficiencies in precision or recall are addressed by examining counterexamples with the SMEs and repeating the above process.

Feature Engineering

Warning: Being a knowledge engineer, working with SMEs to create features, can feel like a Rogerian therapy session.

The Best of Both Worlds

You can offer "the best of both worlds" by employing a three-step approach.

1. Bootstrap candidate features.
2. SMEs validate and annotate the features.
3. "Rinse and repeat": Grade the classification results and repeat steps 1 and 2 if needed.

The Best of Both Worlds

You can offer “the best of both worlds” by employing a three-step approach.

1. You start by bootstrapping the feature-engineering process with candidate feature learning.
    - Possibly use differential phrase-frequency analysis to generate human-friendly features with default weights.

For classifying blog posts as “conservative leaning”:

Machine-generated candidate features:

Phrase	**Score**	**Theme**
2nd amendment	0.56	
RINO	0.76	
Ronald Reagan	0.43	
libtard	0.89	
Nikki Haley	0.77	
welfare state	0.82	
next election	0.33	
revolving door	0.41	

The Best of Both Worlds

You can offer “the best of both worlds” by employing a three-step approach.

1. You start by bootstrapping the feature-engineering process with candidate feature learning.
2. Involve SMEs in validating and/or weighting the candidate features.
   - SMEs can assign themes, ranks, and/or weights to the features they approve.

For classifying blog posts as “conservative leaning”:

SME-annotated features:

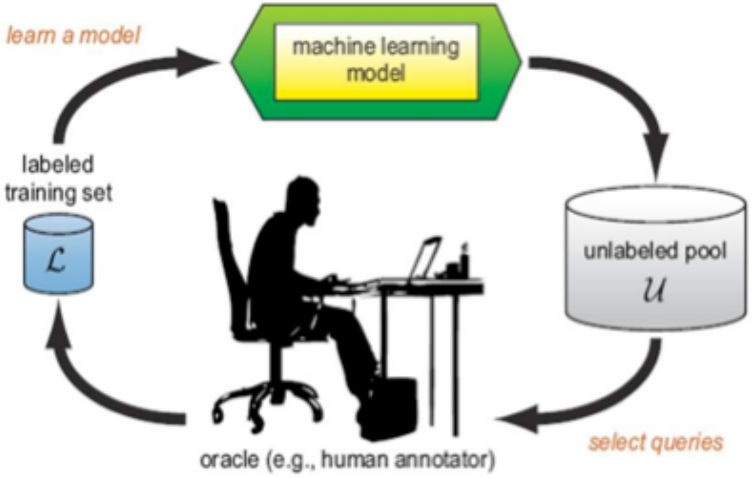
Phrase	**Score**	**Theme**
2nd amendment	0.56	Issues
RINO	0.76	Insults
Ronald Reagan	0.65	Leaders
libtard	0.89	Insults
Nikki Haley	0.43	Leaders
welfare state	0.82	Issues
~~next election~~	0.00	N.A.
~~revolving door~~	0.00	N.A.

The Best of Both Worlds

You can offer "the best of both worlds" by employing a three-step approach.

1. You start by bootstrapping the feature-engineering process with candidate feature learning.
2. Involve SMEs in validating and/or weighting the candidate features.
3. Upon judging the results, you ask the SMEs to supply relevance feedback, which is used to retune the features as needed.
   - You can repeat step 1 and/or 2 above to improve both precision and recall.

Precision: 84%
Recall: 61%



DataScience@SMU

Top-Down vs. Bottom-Up NLP

Natural Language Processing

Choosing Your Approach in NLP

The question we are asking is:

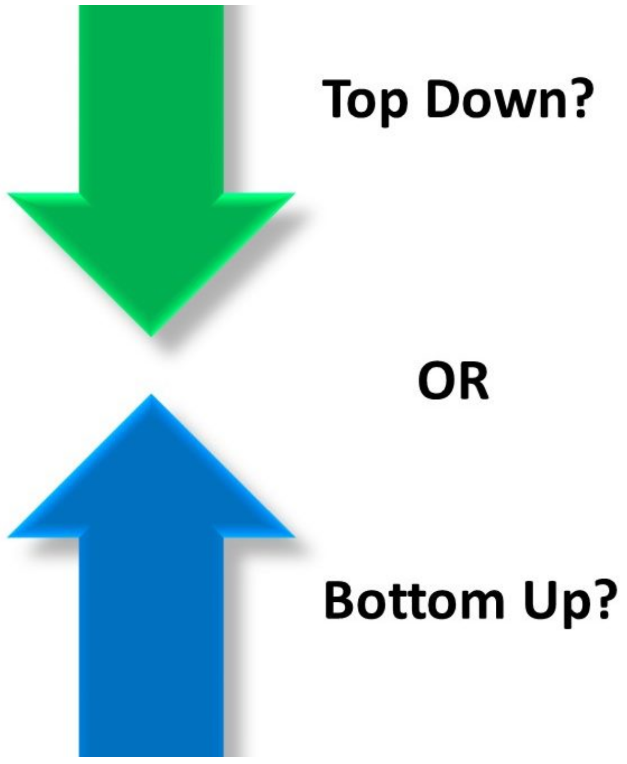
- Shallow vs. deep
- Statistical vs. symbolic
- Feature engineering vs. feature learning
- Top-down vs. bottom-up
  - ***Shall I start a with high-level classifications and text characteristics, then gradually break them down into more detail, or shall I do the reverse?***
- Transparent vs. opaque (AI vs. XAI)

A Difficult Choice

You can build your NLP framework around a set of high-level concepts and categories into which everything fits (a top-down approach)

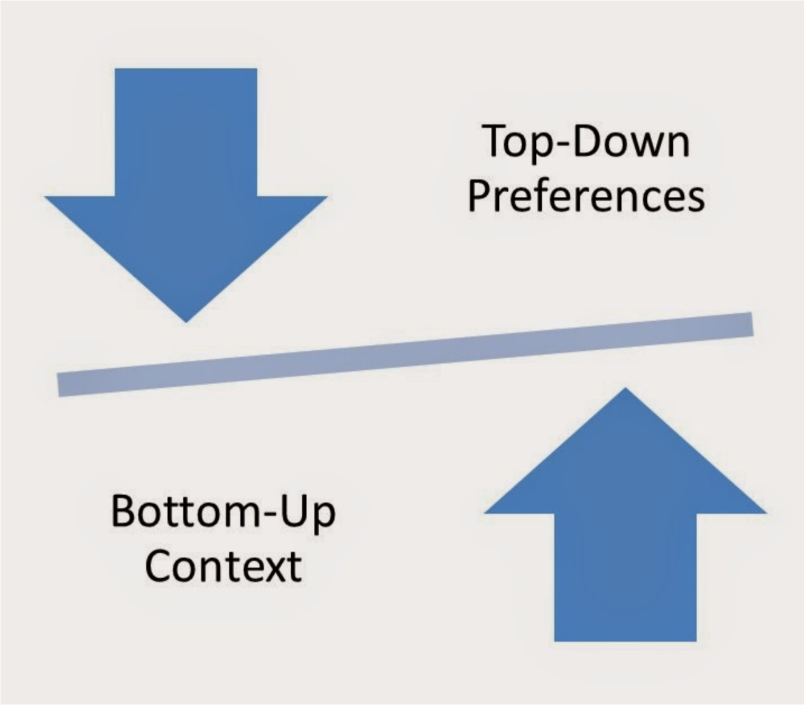
- or -

You can build up a huge list of stemmed keywords from raw text, then try to roll them up into phrases, tag clouds, hypernym trees, and the like (a bottom-up approach).



A Difficult Choice

- Top-down means starting with high-level concepts, such as an upper-level taxonomy.
  - Nontechnical partners love this!
- Bottom-up means starting with detailed NLU (e.g., examining individual word patterns).
  - This sets you up to have rich context information for every feature of each document.



The Top-Down Approach

Examples of the top-down approach are the IAB taxonomy, the Amazon directory, or the site map of any large publisher's website such as Newsweek, WSJ, NYTimes, or CNET.

REVIEWS NEWS VIDEO HOW TO SMART HOME CARS DEALS DOWNLOAD

Best Products	Desktops	Networking	Tablets
Appliances	Drones	Phones	TVs
Audio	Headphones	Printers	VPNs
Cameras	Laptops	Smart Home	Wearables & VR
Cars	Monitors	Software	Web Hosting

The Top-Down Approach

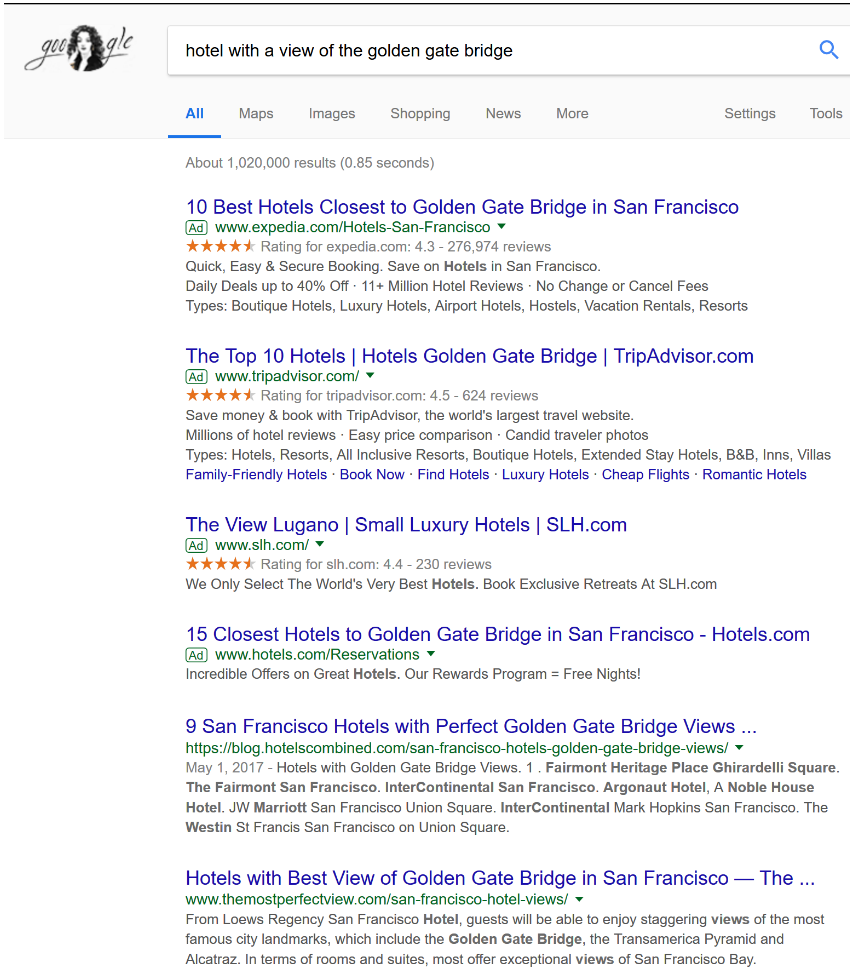
Examples of the top-down approach are the IAB taxonomy, the Amazon directory, or the site map of any large publisher's website such as Newsweek, WSJ, NYTimes or CNET.

PROS:
A neat-and-tidy "category tree" can be used to classify all the content, and the number of categories can be kept down to a manageable number (usually dozens or hundreds of categories).

CONS:
Editors must have decided the categories of content ahead of time and every new item must fit into one or more of those preestablished categories.
Otherwise, a new category must be requested or created ad hoc.

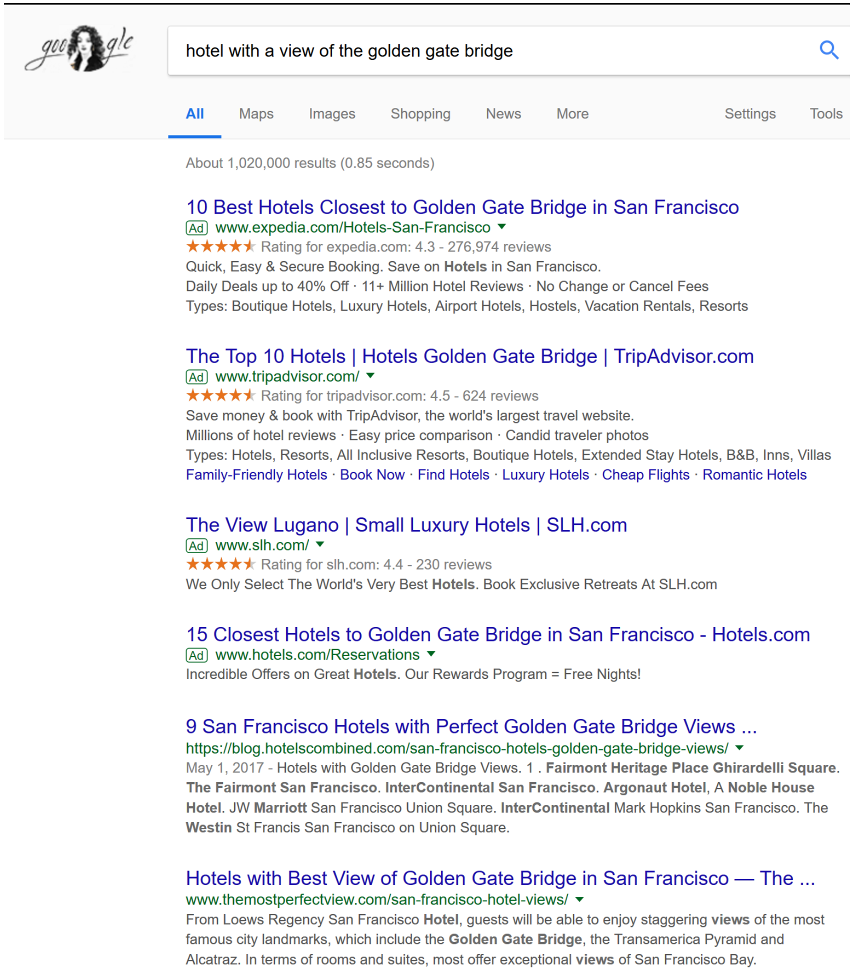
The Bottom-Up Approach

- Google's keyword search index is an example of the bottom-up approach.
- Every word of every document is indexed.
- These keywords, taken en masse, constitute the primary material of the information service.



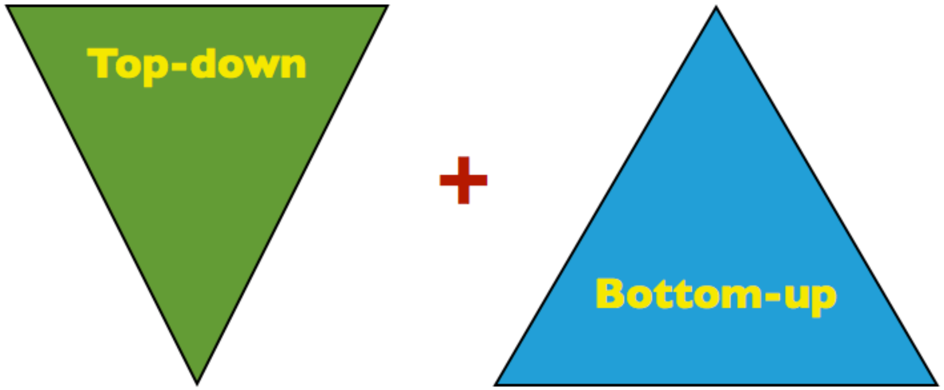
The Bottom-Up Approach

- One advantage of this approach is that newly coined words and phrases (as well as misspelled words) are promptly crawled and indexed.
- If a new rock-n-roll band opens up a website, calling itself “Frontera” (a made-up word), then that word is immediately entered into the index just like any other keyword.



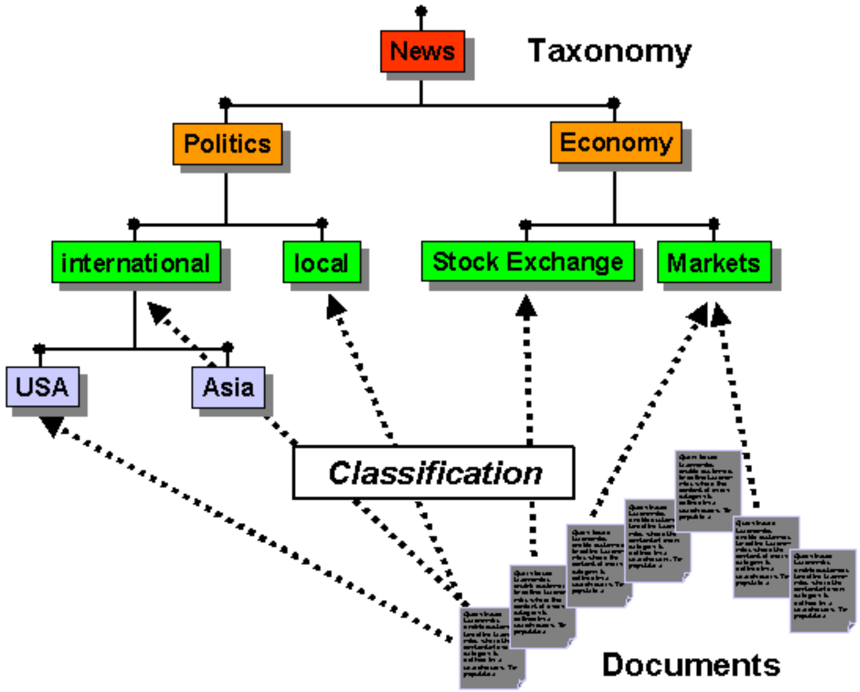
Combining the Approaches?

- Many organizations doing NLP exclusively employ a top-down or bottom-up approach to organizing and understanding their content repository.
- It is possible to blend both approaches.



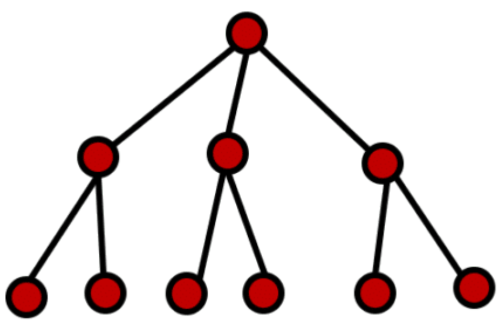
Combining the Approaches

- You can use the top-down approach, e.g., document classification, to control which large subsets of a corpus are of interest for a given use case (e.g., for a given app, site, user, session).
- Thus, for a news collection, you can handle just the economic content, or just the politics, or within politics, just Asia politics content.

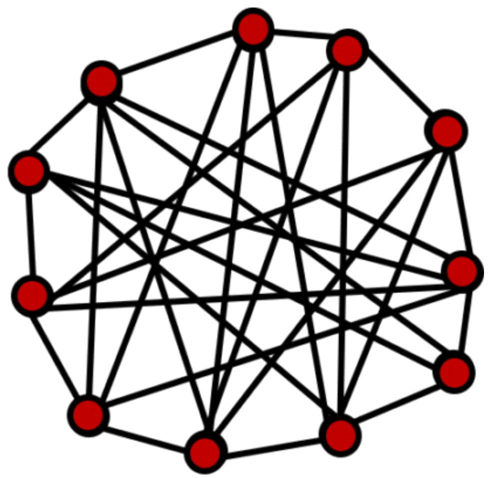


Combining the Approaches

- Once you have taken a subset of your overall corpus, you can turn to your bottom-up method to ensure that you have broad representation of phrases, names, and topics.
- You can then start to cluster or subcategorize those phrases (e.g., NEs or tags or topics).



"Top-down"



"Bottom-up"

A Final Word on Middle Ontology

- Cognitive linguistics teaches us that humans prefer roughly the middle level of a hierarchical ontology.
- This is a critical insight for employing either the top-down or the bottom-up approach to NLP.

Cognitive linguistics is an interdisciplinary effort to combine computer science, linguistics, neuroscience, psychology, anthropology, and philosophy on understanding how humans learn and use language.

A Final Word on Middle Ontology

- To illustrate the innate preference for middle-level ontology, consider this example.
- Suppose you see a dog knock over a trash can in the kitchen. You might say to the friend next to you, “Hey, look at what the dog is doing!”

A Final Word on Middle Ontology

- You are very unlikely to reference a higher-level ontology to say “Hey, look at what that mammal is doing.”
- Nor are you likely to use a low-level ontology to say, “Hey, look at what that standard smooth-haired dachshund is doing!”

A Final Word on Middle Ontology

We naturally default to a level of specification that is neither obtusely abstract nor painstakingly particular. We live at the middle.

This means:

- ***Top-down approaches aren’t very useful until they go at least three or four levels deep from the top.***
- ***Bottom-up approaches aren’t very useful until they go up to the level of metaclusters that have topic labels.***

In either case, you arrive at the middle!

DataScience@SMU

Transparent vs. Opaque NLP

Natural Language Processing

Choosing Your Approach in NLP

The question we are asking is:

- Shallow vs. deep
- Statistical vs. symbolic
- Feature engineering vs. feature learning
- Top-down vs. bottom-up
- Transparent vs. opaque (AI vs. XAI)
  - ***Shall I engineer a system that is explainable to an intelligent SME, or is it okay to build something only my fellow data scientists and engineers could understand?***

Transparent vs. Opaque

- Some services are “opaque” in that there is no way to view, in layman’s terms, an explanation of “why” the system gave a particular result.
  - For example, why was an article classified as sports when I think it should be health?
  - In an opaque system, someone with a PhD may be required to spend several hours tracing through a few thousand data points to unravel why the system arrived at the answer that it did.

Transparent vs. Opaque

- In a transparent system, the reason that the system gave a particular answer would be revealed fairly clearly and easily.
  - Perhaps a rule dictates that “NHL” and “NBA” indicate sports, and these terms were found in the document.
  - Or it could be something else, such as there being several documents of similar characteristics already classified as sports, which were found to be the most strongly comparative examples.
  - Either way, the process is readily understandable, even by a nontechnical person such as an editor, author, or agency representative.

How to Decide

- If you have SMEs as stakeholders and they have clout, then you will want to adopt transparent methods whenever you can—which is nearly all the time.
- In some cases, such as with document classifiers, incredible scale can be accomplished with opaque methods that otherwise would not be feasible with transparent methods.
- So if you need massive scale, the content exhibits frequent model changes, and you don't need to (or can't) interface with strong SMEs, then opaque, ML-driven methods are the way to go.

If You Want to Stay Transparent

- Doing opaque NLP is the easier path these days because ML-driven tools are abundant. Building transparent systems is trickier.
- To stay transparent, make sure your outputs can be traced to lexical entries, encyclopedic entries, reference documents, readable rules, or readily visible statistical metrics.
- Seek to have one of the more “explainable” systems in the AI community.

AI vs. XAI

A word about terminology: recently the term “XAI” (meaning “explainable AI”) has come into vogue—and it simply means transparent rather than opaque AI.

BUZZORD ALERT!
“XAI”

DataScience@SMU