Text Mining and Topic Modeling

Orion Penner

31/30.05.2019

Competition and Innovation Summer School

Ulcinj, Montenegro

The plan

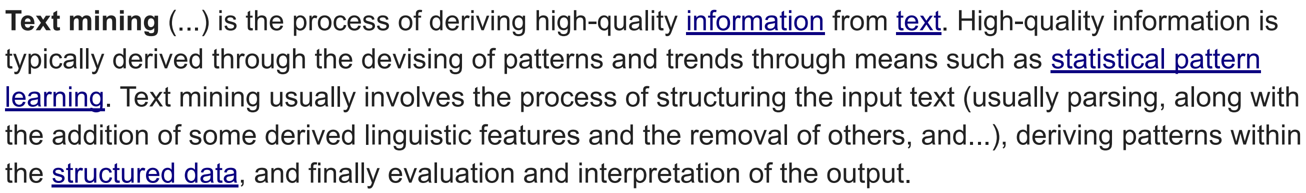
Two sessions:

Foundations of Text Mining

Topic Modeling

What is Text Mining?

From wikipedia:



> **Text mining** (...) is the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. Text mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and...), deriving patterns within the structured data, and finally evaluation and interpretation of the output.

1. Structuring text →Converting it to a mathematicalized form
2. Identifying patterns and trends (and characteristics) through statistical techniques.

What is Text Mining?

What I am not going to cover today are cases in which you are trying to extract or structure specific, **predefined**, data from text.

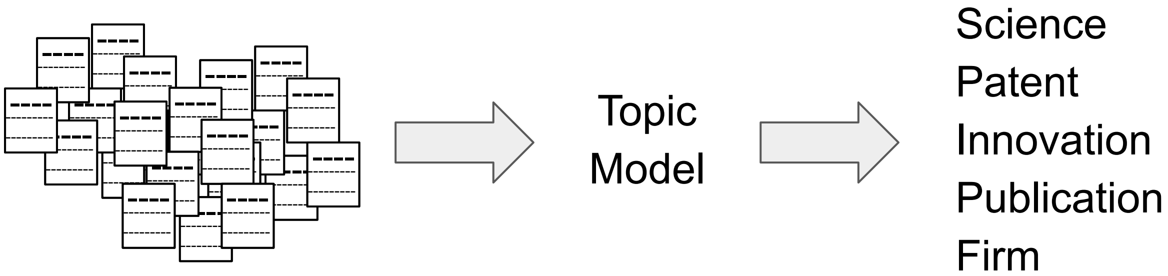
ex.

extracting a city name from an inventor address.
extracting company names from a science park webpage.
extracting a number from a financial report.
extracting patent product pairs from a webpage.

Today I am focused on cases where text **is** the data.

What is Topic Modeling?

Statistical models for extracting the abstract “topics” from a set of documents.



Topic Model

Science

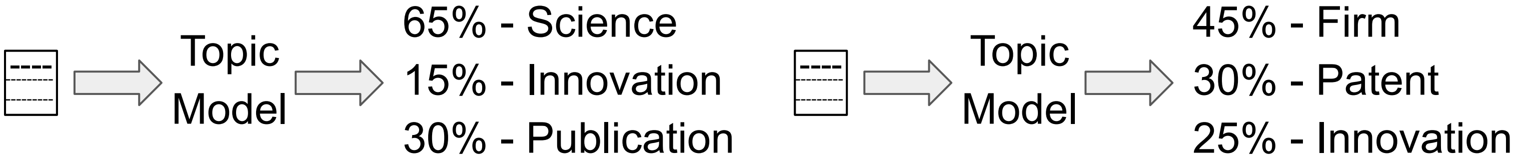
Patent

Innovation

Publication

Firm

But more interestingly, they tell us the relative weight of each topic within each individual document.



Topic Model

65% - Science

15% - Innovation

30% - Publication

Topic Model

45% - Firm

30% - Patent

25% - Innovation

Roadmap

- Motivation
  - Examples from economics
- Definitions
- Pre-processing
- Dictionary Methods
- Vector Space Models

Motivation

Very pragmatic:

- There is an enormous amount of textual data out there.
- It is largely untapped in economics and the social sciences.
- Computational power and techniques are at a point where processing large amounts of text is fairly “easy”.

This data opens avenues to:

1. Measure/estimate/proxy better quantities and variables of long standing interest.
2. Measure/estimate/proxy phenomena previously uncaptured.

Motivation - Caveat

Even if this stuff is “easy” it is not easy.

On the computational side, you have to be able to code (troubleshoot).

On the intuition side, most techniques require a good amount of tacit knowledge to “get right”.

- ➔ It isn’t likely you will find that tacit knowledge in your personal network, so has to be earned through trial and error.

I cannot guarantee what the return on investment would be for you and your career. I am confident text based analysis will play a not minor role in the social science over the next 10-15 years, but I’m biased.

Motivation - Examples

Firm-Level Political Risk: Measurement and Effects

Tarek A. Hassan, Stephan Hollander, Laurence van Lent, Ahmed Tahoun
R&R QJE

Transcripts of earnings conference calls.

- Develop a firm-level measure of political risk.
- Further decompose into 8 subclasses of political risk.
- Do a lot of validation on the measure.
- Cleanest result is likely:

  > "Firms exposed to political risk retrench hiring and investment and actively lobby and donate to politicians."

Motivation - Examples

Firm-Level Political Risk: Measurement and Effects

Tarek A. Hassan, Stephan Hollander, Laurence van Lent, Ahmed Tahoun

R&R QJE

Their measure:

- Bigrams, differentiated into political and non-political.
  - Differentiation based on training on Political Science textbooks and political news articles.
- Then count occurances of political bigrams that are within a distance 10 of synonyms of risk/uncertainty.

Motivation - Examples

Transparency and Deliberation Within the FOMC: A Computational Linguistics Approach

Stephen Hansen, Michael McMahon, Andrea Prat

QJE, 2018

Look at deliberations of the Federal Open Market Committee.

- ➔ Key decisions on interest rate and money supply through control of the Fed's buying and selling of United States Treasury securities.

- Deliberations were recorded and transcribed just to aid with drafting minutes (since 1970's).
- But in 1993 Fed was forced to open all past and future deliberations (with 5 year delay for future).

Motivation - Examples

Transparency and Deliberation Within the FOMC: A Computational Linguistics Approach

Stephen Hansen, Michael McMahon, Andrea Prat

QJE, 2018

DiD around the policy change to see if the new transparency led members:

- Acquire and share more information on the economy. Interpreted under “career concerns” theory as them working harder due to knowledge that history will judge them. (“discipline”)
- Conform more to the prevailing opinion within the room so as to not be at risk of being singled out if wrong.

Motivation - Examples

Transparency and Deliberation Within the FOMC: A Computational Linguistics Approach

Stephen Hansen, Michael McMahon, Andrea Prat

QJE, 2018

Specifically, they used topic modeling (Latent Dirichlet Allocation) with the DiD to see the extent to which transparency increased certain topics and issues.

What they see:

- More economic topic coverage and references to data in early stage deliberations. (they place in line with “discipline” interpretation).
- Less discourse, less economic topic coverage, and stated opinions being closer to Fed Chair (Greenspan) in late stage deliberations. (they place in line with “conformity” interpretation).

Motivation - Examples

Text as Data
Matthew Gentzkow, Bryan T. Kelly, Matt Taddy
forthcoming at JEL

Covers a wide variety of the basics of text analysis from the point of view of modern economics.

Covers a lot/most of what I am covering today.

Motivation - Examples

Text matching to measure patent similarity

Sam Arts, Bruno Cassiman, Juan Carlos Gomez

SMJ, 2018

Focused on estimating the similarity of patents based on text.

Specifically:

- Take title and abstract from USPTO patents 1976-2013.
- Pre-process each to a set of "unique keywords".
- Measure similarity using Jaccard index (only patents filed in same year).

Motivation - Examples

Text matching to measure patent similarity

Sam Arts, Bruno Cassiman, Juan Carlos Gomez

SMJ, 2018

Carry out a great deal of validation, both statistical/algorithmic and also with experts.

Finally, reproduce:

> Thompson P. 2006. Patent citations and the geography of knowledge spillovers: evidence from inventor-and examiner-added citations. The Review of Economics and Statistics 88(2): 383-388.

Motivation - Examples

Text matching to measure patent similarity

Sam Arts, Bruno Cassiman, Juan Carlos Gomez

SMJ, 2018

Similar effort:

> **Patent-to-Patent Similarity: A Vector Space Model**
>
> Kenneth A. Younge and Jeffrey M. Kuhn
>
> Working paper, 2016.

Pure vector based model using more patent text and calculated for every pair of USPTO patents.

Motivation - Examples

Measuring technological innovation over the long run
Bryan Kelly, Dimitris Papanikolaou, Amit Seru, Matt Taddy
Working paper, 2018

Develop a text based similarity measure.

➔ Vector space model, using special version of TF-IDF.

Significance of a patent is its average similarity with patents in the near future (5 years) divided by average similarity with patents in near past (5 years).

Motivation - Examples

Measuring technological innovation over the long run

Bryan Kelly, Dimitris Papanikolaou, Amit Seru, Matt Taddy

Working paper, 2018

Validate in a variety of ways:

- Comparison to citations.
- Overlap with breakthrough patents identified by RAs.
- Tobin Q.

Then use trends in significance measure to identify waves of innovation.

Roadmap

- Motivation
    - Examples from economics
- Definitions
- Pre-processing
- Dictionary Methods
- Vector Space Models

Definitions - Documents

The easiest way to start thinking about documents is with examples, like:

- Patents
- Publications
- Legal documents
- Newspaper articles
- Tweets.

Thus emerge some basic properties of documents:

- They contain text
- They are likely the objects you are trying to characterize through text analysis.

Definitions - Documents

A document is, fundamentally, the object you are trying to infer (from its text) some characteristics of.

But don’t follow into rigid thinking: Documents can be anything, provided you are somehow able to interpret the results. *ex*.

- All patents in a given year and patent class. Thus you are characterizing that area of technology as a whole. Similarly, fields or journals in science.
- But can also disaggregate, then reaggregate as well. For example, take the different sections of scientific publication (introduction, methods, *etc*) in one year and aggregate them into one document and compare overall characteristics of the different sections.

Definitions - Corpus

A corpus is the collection of documents you will analyze. *ex*.

- All USPTO patents.
    - Or all patents in a given year.
        - Or all patents in a given year and technology class.

But keep in mind that when corpus is discussed in a technical manner, it almost always referred to exactly the set of documents that are being fed to the analysis.

Definitions - Bag-of-words

Bag-of-words refers to an approach for representing documents in which only the counts of each word are consider.
i.e. grammar and word order are ignored.

“the dog is black and grey”

Today I will only deal with bag-of-words*.

*word2vec & doc2vec are exceptions.

Word	Count	Word	Count
“the”	1	“black”	1
“dog”	1	“and”	1
“is”	1	“grey”	1

Definitions - Words, Terms and Tokens

Thus far I have lead you to believe that words are the fundamental objects comprising a document.

However, when structuring text (representing it mathematically) this need not be the case.

Individual elements can be more than one word, less than one word (for compound words) or an ontological keyword/code assigned to a word.

Whatever we end up using as our “atoms” of text we refer them to tokens. But sometimes people call them terms. And often (sloppily) words.

Roadmap

- Motivation
    - Examples from economics
- Definitions
- Pre-processing
- Dictionary Methods
- Vector Space Models

Pre-processing

For most applications non-trivial pre-processing is required.

While each stage can be, itself, quite complex the overall goal is quite simple:

> Reduce the textual data to a set of features that will allow your statistical techniques to best capture the content of the text as concerns your goals.

In text mining this feature selection/engineering process is referred to as **tokenization**.

Pre-processing

Today I will cover:

1. Character encodings
2. Case folding
3. Stop words
4. Punctuation
5. Lemmatization
   a. Stemming
6. N-grams

The order of which generally reflects the order in which they should be done.

Character encodings

Character encoding issues will arise most often when dealing with characters arising in languages other than english. *ex*.

- Cyrillic.
- Chinese, Japanese and Hangul characters.
- Accented characters of Slavic, Germanic and Magyar origin.
- *etc*.

The first two require quite different machinery, and probably an approach geared to specific languages.

The problem of accents though, can be sorted.

Character encodings

Handling accents:

If you are very certain that the proper characters will be used throughout your corpus, keep them.

However, it is often the case that in, for example, reporting names simplified versions are provided as often the proper versions.

So generally speaking it is best to just convert to the unaccented version of any character.

Which should be done case by case, and is not fun...

Case folding

Computers interpret words (strings) in a very naive fashion.

The ≠ the

So case folding is the process of “correcting” the case of words such that the computer will interpret them correctly.

Case folding

The most brute force (and common) way of doing this is to simply make every word lowercase.

The → the

Human → human

etc.

But this can induce errors with proper names/nouns.

Bush → bush

In 2002 President Bush… →in 2002 president bush…

The car ran into the bush...→ the car ran into the bush...

Case folding

The most brute force (and common) way of doing this is to simply make every word lowercase.

The → the

Human → human

etc.

But this can induce errors with proper names/nouns.

Bush → bush

In 2002 President Bush… →in 2002 president bush…

The car ran into the bush...→ the car ran into the bush...

Case folding

More nuanced approach is to lowercase words capitalized following a punctuation but preserve uppercase words in the middle of a sentence.

In 2002 President Bush… →in 2002 President Bush…
The car ran into the bush...→ the car ran into the bush...

In this instance this approach obviously does the trick!

Case folding

But take care:

Bush was the 43rd president. →bush was the 43rd president.

i.e. if your proper noun leads a sentence, its case will be dropped.

In 2002 President bush… → in 2002 President bush…

i.e. is extremely sensitive to typos.

ex. Patent assignees

Hitachi Corporation, hitachi corporation, Hitachi Corporation, ...

Case folding - Wrap up

You must always do some case folding.

- If your data is extremely clean, more nuanced approaches may be viable.
- But in the vast majority of cases, simply lowercasing everything is the best option.

Case folding - Wrap up

You must always do some case folding.

- If your data is extremely clean, more nuanced approaches may be viable.
- **But in the vast majority of cases, simply lowercasing everything is the best option.**

Punctuation

The general rule of thumb for punctuation, especially for bag-of-words approaches is just to strip out all of the punctuation.

Things like commas, periods and other sentence ending marks (?! *etc*.)

Apostrophes can also be stripped:

    aren’t →arent

But be careful you aren’t mapping a conjunction onto a proper noun!

Hyphens can actually often be kept, but may depend upon the context in which they are being employed, so know your data.

Stop words

Stop words are words that carry little information due to their ubiquitous nature in a given language.

ex. “a”, “the”, “of”, “I” in English.

i.e. they are just so common that it is unlikely that anything distinctive or characteristic of a given document will arise from them.

Typically in text mining, all of the stop words are just stripped (removed) from the data.

Stop words

Standard stop word lists exist for many, many, languages.

But, when working with specialized text (*ex*. patents, publications, judgements, public financial filings, *etc.*) there is almost certainly going to be additional ubiquitous words.

Patents → ”comprising”

Publications → Discipline dependent *ex*. ”organism”, “cell”

Judgements → ”whereas”, “investigation”

Financial → “assets”, “liabilities”

Stop words - Wrap up

Your process for stop words will almost always be:

1. Get a standard list of stop words for the language you are dealing with.
2. Augment this list with stop words specific to your context.
3. Strip all of them from your corpus.

Although in some cases, like n-grams (coming later) this may not always be your only option.

Lemmatization

Lemmatization is the process of converting each word to its lemma*.

*Its canonical or “dictionary” form

walks, walking, walked → walk

cars, car’s cars’ → car

sees, seeing, saw → see

am, are is → be

Lemmatization

So some cases are easy, others are hard.

In particular, if are going to do something where **verbs** are important it is critical.

But there are always some NLP packages that can work pretty well on “normal” english.

But for technical language you can often have to be a bit careful, and perhaps build a lemmatizer of your own.

Lemmatization

Often you will find cases where lemmatization is:

- Overkill, because can get what you want with less effort
- Impossible (or at extremely difficult) because a lot of the cases you are dealing with are non-standard english.

So **stemming** is often a much more attractive option.

Stemming

Stemming is another approach for reducing a word to its root.

But unlike lemmatization it generally follows a more rules based approach.

Specifically, most focus on suffix-stripping. Removing from the end:

- s
- ing
- ly
- ed
- ...

Stemming

walks, walking, walked → walk

sees, seeing → see

But saw ???

And just forget about am, are is, be

Some big mistakes can also appear:

fly → f

Stemming

So stemming is often a quick and dirty way to prepare some data for analysis, but have to be really careful of false positives (fly → f) and false negatives (saw → saw).

Some of the dumbest cases can be solved with a lookup table, but efficiency gains are quickly lost.

N-grams

An n-gram is a contiguous sequence of *n* words from within a document.

“the cat is black”

2-grams: [“the cat”, “cat is”, “is black”]
3-grams: [“the cat is”, “cat is black”]

In bag-of-words approaches, a term need not be just a word. It can also be an n-gram.

In fact, most techniques are fully valid taking n-grams as the tokens.

N-grams

It is also possible to mix and match n-grams and individual words within the documents and corpus.

This can be useful, for example, when dealing with company names.

A good rule of thumb is when $P(AB) >> P(A)*P(B)$ you should consider treating those two words as a native 2-gram.

But beware that an n-gram representation of your entire corpus will result in the dimension of your term space to explode.

Roadmap

Motivation

  Examples from economics

Definitions

Pre-processing

Dictionary Methods

Vector Space Models

Dictionary Methods

Dictionary methods use a set of keywords defined by the user to characterize the text contained in each document.

The most common example of a dictionary method is sentiment analysis.

In sentiment analysis, the frequency of words of “positive” and “negative” sentiment within a given document are used to estimate its overall sentiment.

Dictionary Methods

So as an example let’s consider the two tweets:

Back from Japan after a very successful trip. Big progress on MANY fronts. A great country with a wonderful leader in Prime Minister Abe!

So this tweet would score +4 in a simple sentiment analysis

....Super Predator was the term associated with the 1994 Crime Bill that Sleepy Joe Biden was so heavily involved in passing. That was a dark period in American History, but has Sleepy Joe apologized? No!

While this would score -1.

Dictionary Methods

Dictionary methods are often enhanced by:

- Increasing the weight of words that fit better the targets.
  - *ex*. in sentiment analysis “stupendous” would count more than “good”.
- Similarly, rare words can be weighted higher than common words.
  - *ex*. In sentiment analysis, “balk” would count more negative then “bad”. The idea being that even if a word is not more forcefully negative/positive, by being more rare it is more specific, and more likely to be truly negative/positive.

Dictionary Methods - Co-occurrence

To make something interesting of dictionary analysis it is often necessary add some co-occurrence to the analysis.

For example, the relationship between the sentiment of the tweet and the country or person that features most prominently in it.

Or within a patent, the co-occurrence of a specific “use” keyword and a “technology” keyword.

Dictionary Methods - Limitations

Dictionary methods have to easily identified weakness:

1. In different domains, words clearly have different meanings.
2. Local context of a given word is often key to its meaning.
3. Validation is difficult.
4. Significant domain knowledge is required and efforts from other (even adjacent) domains are often not useful.

Dictionary Methods - Limitations

Domain differences

In different domains words obviously carry different meaning.

ex. Table → A physical table.

Table → A row-column structure within a database.

Local context

Particularly in sentiment analysis, the meaning of a keyword can be easily inverted by its context:

....Super Predator was the term associated with the 1994 Crime Bill that Sleepy Joe Biden was so heavily involved in passing. That was a dark period in American History, but has Sleepy Joe apologized? No!

Dictionary Methods - Limitations

Validation

There is no clear cut framework for validating a dictionary.

Solutions tend to be either brute force (RAs) or *ad hoc* (exploiting the basic distributions of the system).

Domain knowledge

In light of the above concerns, substantial domain knowledge is required to build an appropriate dictionary.

This is often the biggest issue, but for social scientists it is much less problematic.

Dictionary Methods - Wrap up

Dictionary methods were, for a long time, really the only reasonable option for text mining in the social science due to lack of techniques and computing power.

They have some prominent flaws. Particularly as concerns context.

But can still be very useful if employed by someone with substantial domain knowledge and a corpus that sticks to that domain.

Roadmap

Motivation

Examples from economics

Definitions

Pre-processing

Dictionary Methods

Vector Space Models

Vector Space Models

In Vector Space Models each document is represented by a vector, in which each element corresponds to how frequent a specific term is within the document.

Document *i* = “the cat is black”
Document *j* = “the dog is black and grey”

	“the”	“cat”	“is”	“black”	“dog”	“and”	“grey”
d_i	1	1	1	1	0	0	0
d_j	1	0	1	1	1	1	1

Vector Space Models

The purpose of a vector space model is (almost always) to make it possible to compare documents.

In particular, to estimate their similarity or distance from each other.

There are many ways to estimate similarly but the most common by far is cosine similarity.

$$\text{cosine } \theta_{i,j} = \frac{\mathbf{d}_i \bullet \mathbf{d}_j}{\|\mathbf{d}_i\| \, \|\mathbf{d}_j\|}$$

Vector Space Models

So in a corpus of *N* documents, overall comprising *M* unique terms.

Each document is represented by a vector of length *M*.

And the *document-term* matrix is an *N* x *M* matrix.

	“the”	“cat”	“is”	“black”	“dog”	“and”	“grey”
d_i	1	1	1	1	0	0	0
d_j	1	0	1	1	1	1	1

Across all vector space models this general formalism does not change.

But there are many ways for arriving at the term weights.

Vector Space Models - Boolean

In a **boolean** vector space model the values of a document-term vector take the values:

true - if the term is present within the document

false - if the term is not present within the document

	“the”	“cat”	“is”	“black”	“dog”	“and”	“grey”
d_i	true	true	true	true	false	false	false
d_j	true	false	true	true	true	true	true

Here cosine similarity doesn’t work, instead use Jaccard.

Vector Space Models - Count

In a **count** vector space model the values of the document-term vector are equal to the number of times each term appears within the document.

	“the”	“cat”	“is”	“black”	“dog”	“and”	“grey”
d_i	1	1	1	1	0	0	0
d_j	1	0	1	1	1	1	1

Vector Space Models - Text Frequency

In a **text frequency** vector space model the values of the document-term vector are equal to the **relative frequency** of the word within that document.

	“the”	“cat”	“is”	“black”	“dog”	“and”	“grey”
d_i	0.25	0.25	0.25	0.25	0	0	0
d_j	0.167	0	0.167	0.167	0.167	0.167	0.167

Vector Space Models - TFIDF

TFIDF stands for **text frequency, inverse document frequency**.

In TFIDF the values of the document-term vector are equal to the relative frequency of the word within that document multiplied by the inverse document (relative) frequency.

This increases the weight assigned to words that appear in fewer documents (*i.e.* rare words)

And (drastically) decreases the weight assigned to words appearing in many documents (*i.e.* ubiquitous or common words).

Vector Space Models - TFIDF

	“the”	“cat”	“is”	“black”	“dog”	“and”	“grey”
d_i	0.125	0.25	0.125	0.125	0	0	0
d_j	0.083	0	0.083	0.083	0.167	0.167	0.167

Vector Space Models - Strengths

Vector space models generally perform well assuming your data is largish (thousands of documents, with a fair bit of text each).

TFIDF, in particular, is extremely powerful. Sometimes it seems to work almost by “magic”. Its ability to downweight the most common terms makes the analysis much less sensitive to unforeseen stop words.

Considering only bag-of-words techniques that do not make use of a latent space, at TFIDF vector space model is about as powerful a technique as you can find.

Vector Space Models - Strengths

Vector space models generally perform well assuming your data is largish (thousands of documents, with a fair bit of text each).

TFIDF, in particular, is extremely powerful. Sometimes it seems to work almost by “magic”. Its ability to downweight the most common terms makes the analysis much less sensitive to unforeseen stop words.

Considering only **bag-of-words** techniques that do not make use of a **latent space**, at TFIDF vector space model is about as powerful a technique as you can find.

Vector Space Models - Weaknesses

Sticking to **bag-of-words** still means that the local (in sentence) context is lost.

Lacking a **latent space** means that domain specific context will be lost.

> “a **table** comprised of a flat surface and four legs.”
> “a **table** comprised of rows and columns.”

These are both the same dimension in our high dimensional space.

Having to calculate pairwise similarities does not scale well with corpus size.

Vector Space Models - Wrap up

Vector space models represent a nice position in the tradeoff between power and complexity.

Which is probably why they are one of the approaches you’re most likely to come across in economics, and especially in the patent analysis literature.

But lacking latent space representation, a given word/term/token means the same regardless of context.