

# Working with text in Gephi

Clément Levallois

2017-02-28

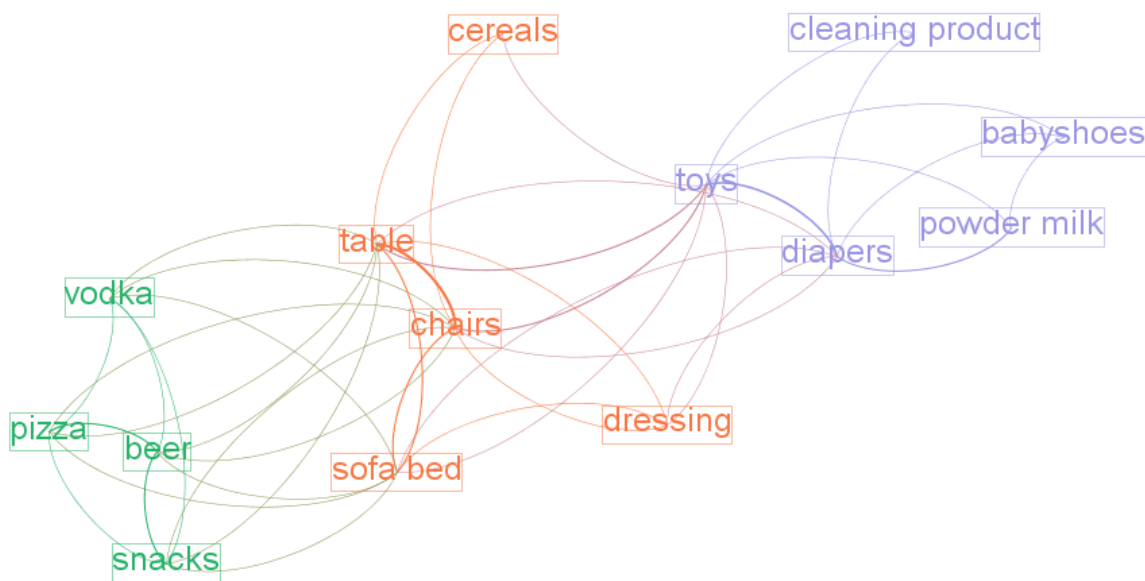
# Table of Contents

Presentation of this tutorial.....	1
Why semantic networks? .....	2
Choosing what a "term" is in a semantic network .....	2
1. Removing "stopwords" .....	3
2. Considering "n-grams" .....	3
2 bis. Considering "noun phrases" .....	3
3. Stemming and lemmatization .....	3
Should we represent all terms in a semantic network? .....	4
1. Start with: how many words can fit in your visualization? .....	4
2. Representing only the most frequent terms .....	4
(to be continued) .....	5
More tutorials on importing data to Gephi .....	5
the end .....	5

# Gephi

## Presentation of this tutorial

This tutorial explains how to draw "semantic networks" like this one:



*Figure 1. a semantic network*

We call "semantic network" a visualization where textual items (words, expressions) are connected to each others, like above.

We will see in turn:

- why are semantic networks interesting
- how to create a semantic network
- tips and tricks to visualize semantic networks in the best possible way in Gephi

# Why semantic networks?

A text, or many texts, can be hard to summarize.

Drawing a semantic network highlights what are the most frequent terms, how they relate to each other, and reveal the different groups or "clusters" of they form.

Often, a cluster of terms characterizes a topic. Hence, converting a text into a semantic network helps detecting topics in the text, from micro-topics to the general themes discussed in the documents.

Semantic networks are regular networks, where:

- nodes are words ("USA") or groups of words ("United States of America")
- relations are, usually, signifying a co-occurrences: two words are connected if they co-occur.

It means that if you have a textual network, you can visualize it with Gephi just like any other network. Yet, not everything is the same, and this tutorial provides tips and tricks on why textual data can be a bit different than other data.

## Choosing what a "term" is in a semantic network

The starting point can be: a term is a single word. So in this sentence, we would have 7 terms:

My sister lives in the United States (7 words -> 7 terms)

This means that each single term is a meaningful semantic unit.

This approach is simple but not great. Look again at the sentence:

My sister lives in the United States

1. **My**, **in**, **the** are frequent terms which have no special significance: they should probably be discarded
2. **United** and **States** are meaningful separately, but here they should probably be considered together: **United States**
3. **lives** is the conjugated form of the verb **to live**. In a network, it would make sense to regroup **live**, **lives** and **lived** as one single node.

Analysts, facing each of these issues, have imagined several solutions:

## 1. Removing "stopwords"

To remove these little terms without informational value, the most basic approach is to keep a list of them, and remove any word from the text which belongs to this list.

You can find a list of these useless terms in many languages, called "stopwords", [on this website](#).

## 2. Considering "n-grams"

So, **United States** should probably be a meaningful unit, not just **United** and **States**. Because **United States** is composed of 2 terms, it is called a "bi-gram".

Trigrams are interesting as well obviously (eg, **chocolate ice cream**).

People often stop there, but I find quadrigrams can be meaningful as well, if less frequent: **United States of America**, **functional magnetic resonance imaging**, **The New York Times**, etc.

Many tools exist to extract n-grams from texts, for example [these programs which are under a free license](#).

## 2 bis. Considering "noun phrases"

Another approach to go beyond single word terms (**United**, **States**) takes a different approach than n-grams. It says:

"delete all in the text except for all groups of words made of nouns and adjectives, ending by a noun"

→ (these are called, a bit improperly, "noun phrases")

Take **United States**: it is a noun (**States**) preceded by an adjective (**United**). It will be considered as a valid term.

This approach is interesting (implemented for example in the software [Vosviewer](#)), but it has drawbacks:

- you need to detect adjectives and nouns in your text. This is very language dependent, and slow for large corpora.
- what about verbs, and noun phrases comprising non adjectives, such as "United States **of** America"? These are not included in the network.

## 3. Stemming and lemmatization

**live**, **lives**, **lived**: in a semantic network, it is probably useless to have 3 nodes, one for each of these 3 forms of the same root.

- Stemming consists in chopping the end of the words, so that here, we would have only **live**.
- Lemmatization is the same, but in a more subtle way: it takes grammar into account. So, "good" and "better" would be reduced to "good" because there is the same basic semantic unit behind

these two words, even if their lettering differ completely.

A tool performing lemmatization is [TextGrid](#). It has many functions for textual analysis, and lemmatization is [explained there](#).

# Should we represent all terms in a semantic network?

We have seen that some words are more interesting than others in a corpus:

- stopwords should be removed,
- some varieties of words (**lived**, **lives**) could be grouped together (**live**).
- sequences of words (**baby phone**) can be added because they mean more than their words taken separately (**baby**, **phone**)

Once this is done, we have transformed the text into plenty of words to represent. Should they all be included in the network?

Imagine we have a word appearing just once, in a single footnote of a text long of 2,000 pages. Should this word appear? Probably not.

Which rule to apply to keep or leave out a word?

## 1. Start with: how many words can fit in your visualization?

A starting point can be the number of words you would like to see on a visualization. **A ball park figure is 300 words max:**

- it already fills in all the space of a computer screen.
- 300 words provides enough information to allow micro-topics of a text to be distinguished

More words can be crammed in, but in this case the viewer would have to take time zooming in and out, panning to explore the visualization. The viewer transforms into an analyst, instead of a regular reader.

## 2. Representing only the most frequent terms

If ~ 300 words would fit in the visualization of the network, and the text you start with contains 5,000 different words: which 300 words should be selected?

To visualize the semantic network **for a long, single text** the straightforward approach consists in picking the 300 most frequent words (or n-grams, see above).

In the case of a collection of texts to visualize (several documents instead of one), two possibilities:

1. Either you also take the most frequent terms across these documents, like before
2. Or you can apply a more subtle rule called "tf-idf", detailed below.

The idea with tf-idf is that terms which appear in all documents are not interesting, because they are so ubiquitous.

Example: you retrieve all the webpages mentioning the word **Gephi**, and then want to visualize the semantic network of the texts contained in these webpages.

→ by definition, all these webpages will mention Gephi, so Gephi will probably be the most frequent term.

→ so your network will end up with a node "Gephi" connected to many other terms, but you actually knew that. Boring.

→ terms used in all web pages are less interesting to you than terms which are used frequently, but not uniformly accross webpages.

Applying the tf-idf correction will highlight terms which are frequently used within some texts, but not used in many texts.

(to go further, here is a webpage giving a simple example: <http://www.tfidf.com/>)

Should you visualize the most frequent words in your corpus, or the words which rank highest according to tf-idf?

Both are interesting, as they show a different info. I'd suggest that the simple frequency count is easier to interpret.

tf-idf can be left for specialists of the textual data under consideration, after they have been presented with the simple frequency count version.

## (to be continued)

## More tutorials on importing data to Gephi

- [The Gephi wiki on importing csv](#)
- [Video "How to import a CSV into Gephi" by Jen Golbeck](#)

## the end

Visit [the Gephi group on Facebook](#) to get help,

or visit [the website for more tutorials](#)