Automatic Extraction and Visualization of Subject-Verb-Object (SVO) Triplets

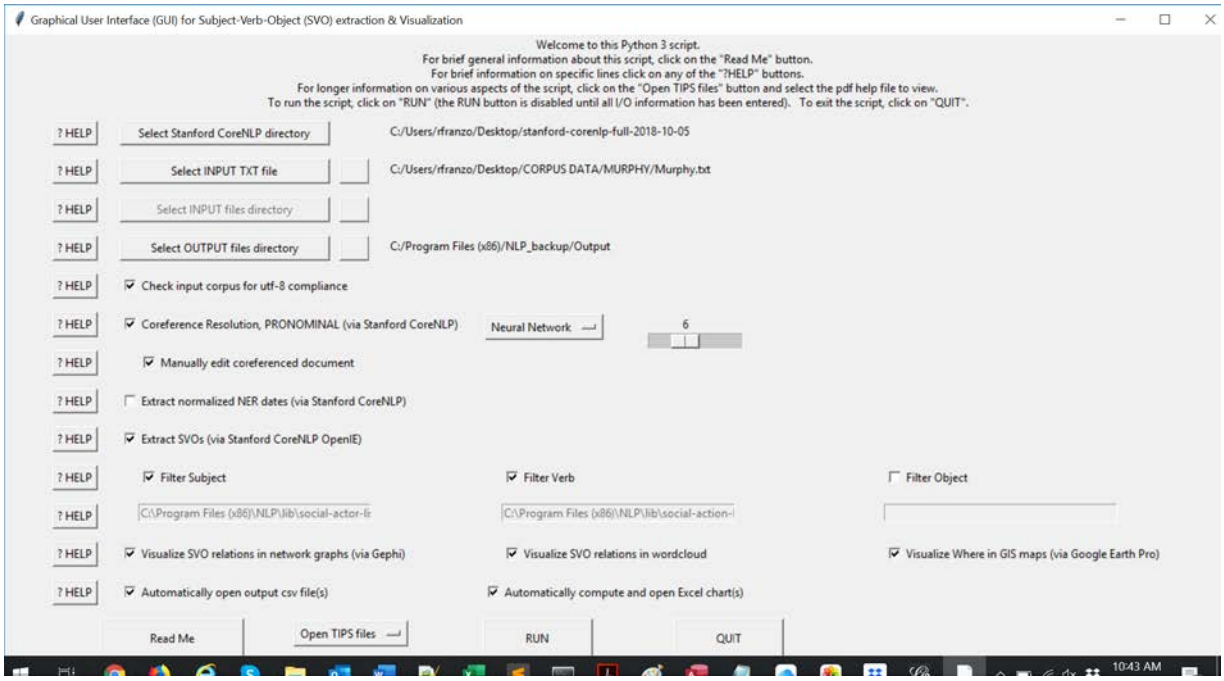
The 5 Ws of narrative: Who, What, Where, When, and Why

The SVO extraction and visualization pipeline is fundamentally about extraction at least some of the 5Ws of narrative: Who, What, Where, When. The SVO are about **Who-What-Whom**, visualized in network graphs (via Gephi). The **Where** is about the locations mentioned in the text and visualized as points on a map (via Google Earth Pro). The **When** is about the times and dates mentioned in the text and visualized as Excel charts.

As for the **Why**… it is beyond the scope of the current pipeline or, more generally, of automatic computational tools.

A convenient GUI for SVO extraction and visualization

The front-end Python 3 scripts provide a convenient and easy-to-use Graphical User Interface (GUI) where the steps required to extract and visualize SVO triplets are clearly laid out.



A pipeline of Java and Python scripts

The pipeline for extracting and visualizing SVO triplets consists of a combination of Python and Java scripts.

In **Python** are written

1. the two SVO_main and SVO_GUI scripts;
2. the utf-8 compliance algorithm that checks for utf-8 encoding of the input file(s), as required by the Stanford CoreNLP tools;
3. the Stanford CoreNLP normalized date extractor;
4. the geocoder and Google Earth Pro;
5. the Gephi dynamic network graphs;
6. the wordcloud of SVO values;
7. the Excel charts.

In **Java** are written several algorithms

1. the Stanford CoreNLP coreference resolution;
2. the basic Stanford CoreNLP parser OpenIE.

utf-8 compliance

Most CoreNLP algorithms require in input text files that are utf-8 compliant, i.e., texts encoded in utf-8 format (on text encoding, see TIPS_NLP_Text encoding.pdf). Make sure to test your input file(s).

Stanford CoreNLP coreference resolution

SVO triplets can represent meaningful sentences in a simplified 3-term structure. In order to improve the quality of SVOs, users can choose to run the pronominal coreference resolution, based on Stanford CoreNLP, before SVO extraction. Basically, coreference would replace pronouns such as *he, she, they, us, …* to their respective nouns. "President Obama said that he would…" would be *resolved* as "President Obama said that Obama would…". To learn more about coreference resolution read TIPS_NLP_Stanford CoreNLP coreference resolution.pdf

Dealing with an imperfect tool: Manual coreference

Automatic coreference is far from a perfect tool with only about 65% accuracy. You can always manually edit the coreference text. **Unfortunately, this option requires a large memory since it brings in memory both files, original and coreferenced side-by-side. A memory hog! You may not have enough memory on your computer for this option.**

Stanford CoreNLP date extractor

The CoreNLP date extractor extracts any temporal expression mentioned in the text in standardized, uniform ways ("normalized dates") that can be compared, sorted, and analyzed to bring out a text's narrative strategy (to learn more about CoreNLP date extractor, normalized dates, and story vs plot narrative strategies, read TIPS_NLP_Stanford CoreNLP date extractor.pdf).

Stanford CoreNLP OpenIE: The engine of the pipeline

At the heart of the SVO approach is the Stanford CoreNLP OpenIE, a java script that is the real engine of the approach. What is OpenIE? Do take a look at the more extensive TIPS TIPS_NLP_Stanford CoreNLP OpenIE.pdf for more in-depth information. But, basically, OpenIE is an Information Extraction tool by the Stanford CoreNLP group that extracts sets of triplets from a text.

Filtering SVOs via dictionary files

OpenIE produces in output triplets of various types. As a social scientist you may be interested in Who Does What, in social actors and social action. No point, perhaps, knowing that a door is blue or made of wood. The SVO GUI provides a way for you to filter each of the three elements of the SVO triplet, keeping only those values for each contained in a supplied dictionary file.

Where do these dictionary files come from?

The NLP Suite comes with three default dictionary files in csv format and stored in the lib subdirectory: the social-actor-list.csv and social-action-list.csv. These files have been constructed using the WordNet tool (Zoom IN/DOWN option with 'Person' as the keyword (synset) starting point. These produce exhausting lists of over 18,000 social actors and 10,000 social actions. For restricted domains these lists may provide both too few and too many entries. For instance, suppose you are working with folk tales. You will find there many talking foxes, flying horses, and many other anthropomorphized animals (e.g., *The Three*

Little Pigs English folk tale in our default sample texts). None of the animals will be in the default social-actor-list.csv. You may wish to build your own list.

How do you build your own filter dictionary lists?

You can build your lists, based on your specific corpus, by running the Stanford CoreNLP to produce the CoNLL table, run an SQL query to obtain a frequency distribution of all tokens (found in the field FORM) that have specific POSTAG labels (e.g., nsubj (nominal subject) or specific NER values (e.g., PERSON), then manually inspect the query results keeping social actors only. You can also use the results of these queries in conjunction with the WordNet tool to see which nouns are animals, for instance.

Input

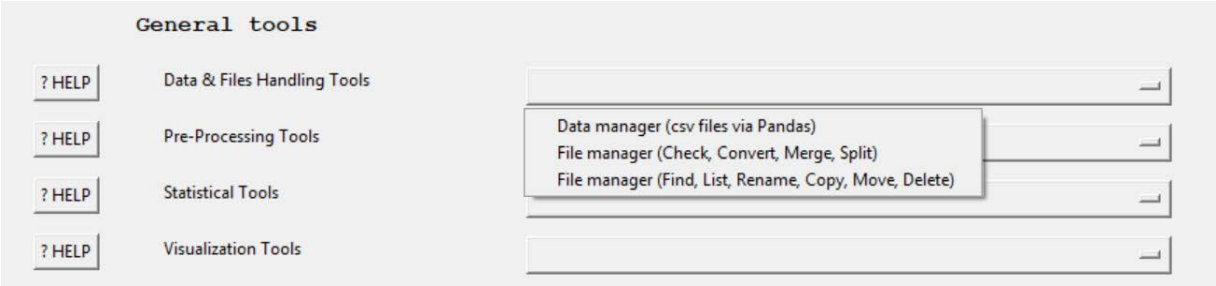
In INPUT the routine expects a text file or a set of text files in an input directory.

Using a merged file in input

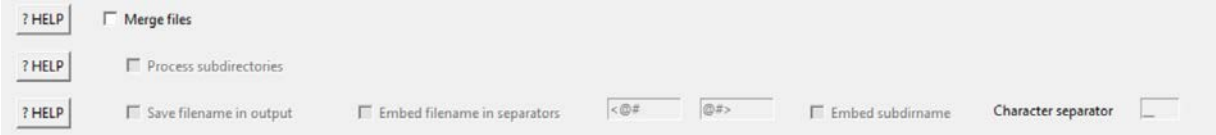
Another analysis strategy for many files, instead of the input directory option where all the text files present in the directory are processed one at a time, is to merge the files into a single file and then process the merged file as a single file.

How do you merge files?

When you run in command line python NLP_main.py, under General tools, Data & File Handling Tools, select the option File manager (Check, Convert, Merge, Split) to open the File manager GUI.



You can also run directly in command line python file_manager_1_main.py to open that same GUI.

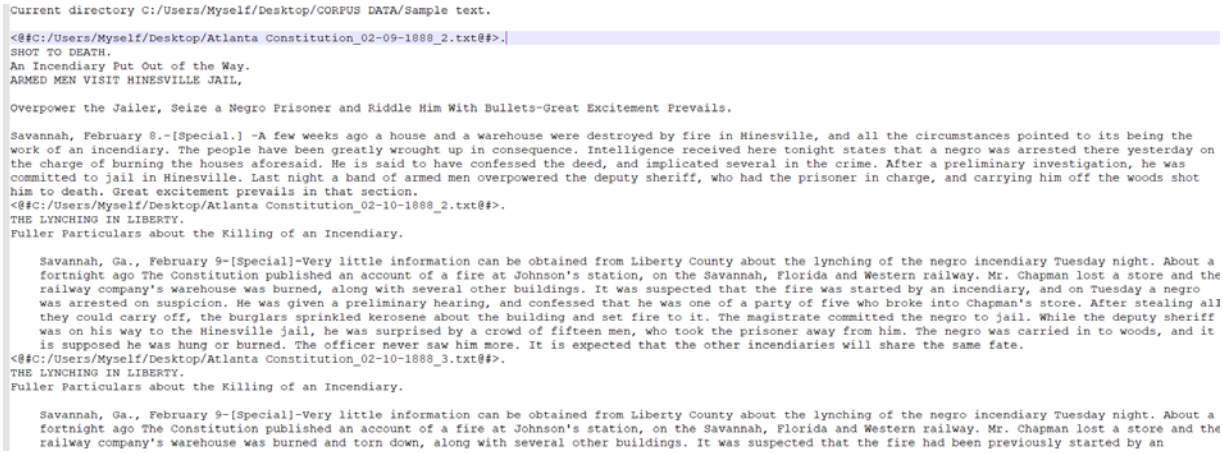


The *Merge files* widget, when ticked, provides many options. You can merge files not just in the main input directory but also in all the subdirectories. You can *Save filename in output*, so as to have a record of each file in the merged file. And to make it easy to search and find you can also embed each processed input filename in special strings, for example <@# and @#>. This way, by searching <@# you will find every filename in the merged file.

Whether files are saved embedding them in special strings or not, **files will be saved with the full directory path** (e.g. <@#C:/Users/Myself/Desktop/CORPUS DATA/Sample text\Atlanta Constitution_02-09-1888_2.txt@#>.). When processing subdirectories, you also have the option to save the filename with an appended suffix: the final part of the subdirectory, separated by the filename with special characters (e.g., __). When using this option, **files are saved without their full path, just the filename** (e.g., <@#Atlanta Constitution_02-09-1888_2__4.txt@#>.) where 4 refers to a specific event/individual/organization ID.

What does a merged file look like?

When saving files with their filenames embed in the default strings, this is what the output will look like using the three sample newspaper articles.



Why would you merge files?

The answer to that question depends upon the NLP tools you are planning to use to analyze your corpus.

1. Some NLP algorithms require in input a set of files, rather than an individual file. Such are Gensim or Mallet topic modelling tools, the NGrams_CoOccurrences tool, or the Shape of Stories tool. **Merging a set of files into a single file will not do you any good if you are planning to use one of these algorithms.** But most tools in the NLP Suite can process indifferently a single file or a set of files.
2. Some NLP tools when processing a set of files in an input directory produce a single output file with clearly marked document ID and document names for each document. Such are, for instance, the Stanford CoreNLP parser or the Python wordcloud tool that will produce images for individua files and the merged file. **Again, no point merging files in these cases.**
3. When processing a set of files in an input directory most algorithms produce results separate for each individual input file. Yet, it may be beneficial to know the results for the corpus as a whole. Such are, for instance, WordNet, the DBpedia/dictionary Annotator, or any other tool where synthetic results are meaningful. **In these case, merge, by all means!**

Output

In output, the SVO script produces several different files, a csv file, a gexf Gephi file, a kml

Google Earth Pro file.

csv files

The main csv file (characterized by the substring _svo in the filename) contains the basic information produced by the SVO script, namely the CoreNLP OpenIE, perhaps with filetered values. Here is what that file looks like using Murphy's *Six miracles* default story.

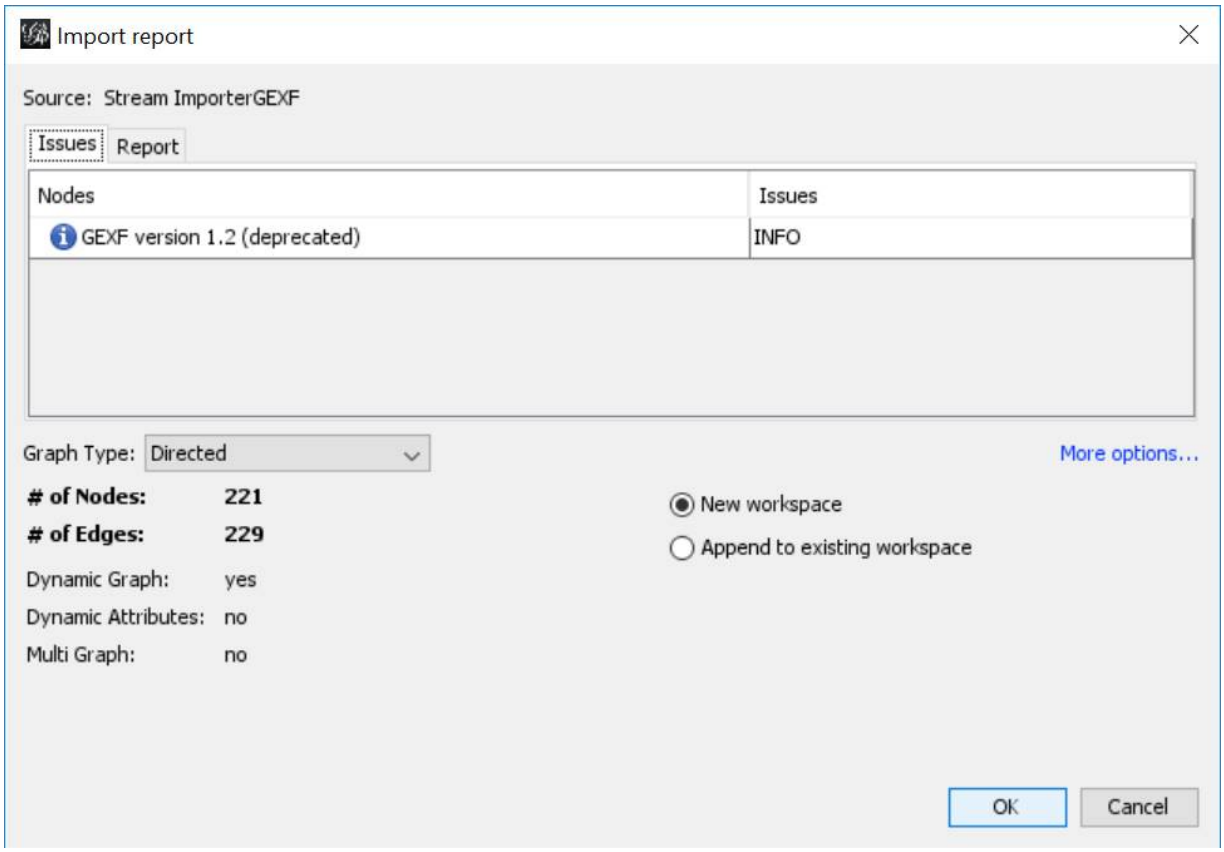
	Sentence	S	V	O/A	LOCATION	PERSON	TIME	TIME_STA	Sentence
2	2	It	was	day			the day af	2019-12-	It was the day after Christmas.
3	3	rows	looming	hospital corridor			this day 4	2065-05-	As I was wheeled backward through the glaring hospital corridor, rows of overhead lights looming up behind me like lit
4	3	rows	looming	me			this day 4	2065-05-	As I was wheeled backward through the glaring hospital corridor, rows of overhead lights looming up behind me like lit
5	3	rows	looming	suns			this day 4	2065-05-	As I was wheeled backward through the glaring hospital corridor, rows of overhead lights looming up behind me like lit
6	5	anesthetis	had warn	me					The anesthetist had warned me that the operating room would feel cold.
7	6	He	introduce	himself					He came to introduce himself to me in the little cubicle where I lay on the gurney.
8	6	I	lay	gurney					He came to introduce himself to me in the little cubicle where I lay on the gurney.
9	7	wife	held	hand		Kathleen			My wife, Kathleen, held my hand and tried to look calm.
10	8	corridor	felt	hospital gown					But even the corridor already felt cold in my hospital gown.
11	9	Everything	was	motion					Everything was in slow motion.

Other csv files provide various information produced, for instance, by the GIS scripts.

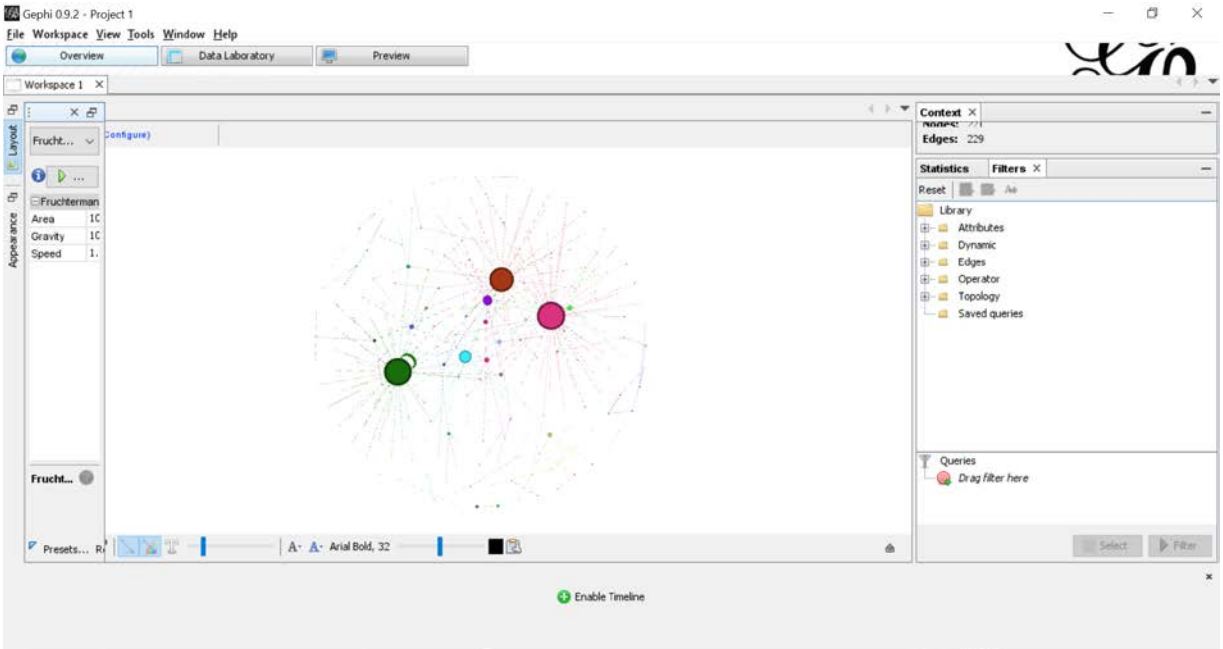
SVO relations: Network graphs in Gephi

Relations between S and O via V can be visualized as network graphs using the freeware Gephi tool. The NLP Suite wrapper for Gephi will automatically construct a dynamic network graph, i.e., a graph that changes overtime and where the sentence index of each SVO is taken as proxy for time. This way, you can see the movement of SVOs across the document.

This is what Gephi looks like when it opens up on the SVO output generated using one of the default documents, Murphy's *Six miracles* story.



Not a helpful visual! Don't be turned off. Easy to bring to life this blob but… please, read all the TIPS files on how to get going with Gephi.

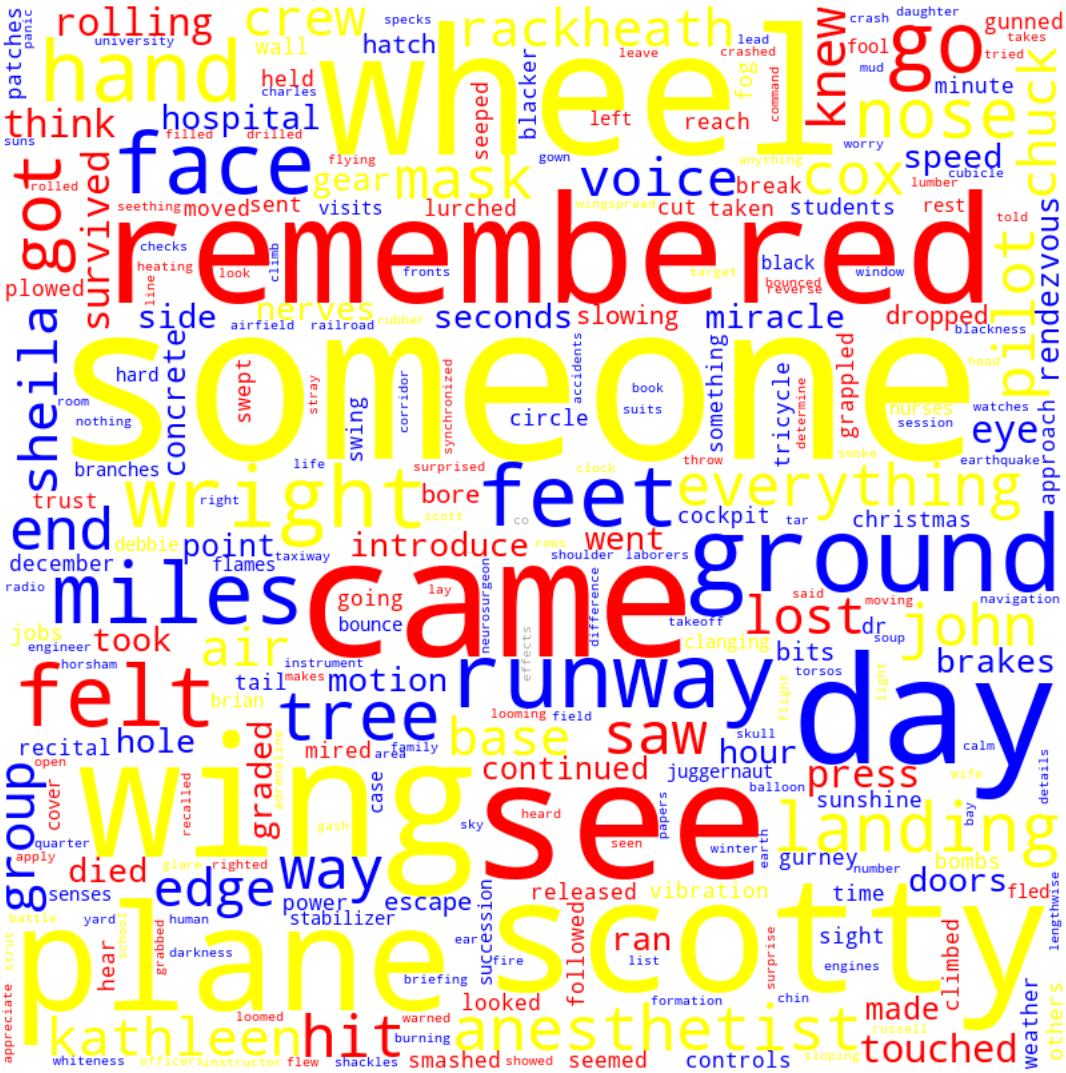


SVO relations: A wordcloud of Subjects, Verbs, and Objects

Another way to visualize relations between SVOs is to map S, V, and O values in different colors in a wordcloud.

Word size and color

The combination of word sizes and colors may hopefully give you a visual sense of what is going on, where **word size** is related to word frequency; and the **colors** indicate S, or V, or O: yellow for S, red for V, blue for O.



Mapping the WHERE: GIS maps in Google Earth Pro

When the Visualize Where option is ticked, the SVO script will automatically geocode any location found in the text and map in using Google Earth Pro. When running SVO using one of the default documents, Murphy's *Six miracles* story, this is what the map will look like.

Excel charts

The SVO pipeline also produces several kinds of Excel bar and line charts to help the user to make sense of the data.

References

Klein, Dan and Christopher D. Manning. 2003. "Accurate Unlexicalized Parsing." *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423-430.

TIPS_NLP_Text encoding.pdf
TIPS_NLP_Stanford CoreNLP coreference resolution.pdf
TIPS_NLP_Stanford CoreNLP date extractor.pdf
TIPS_NLP_Stanford CoreNLP OpenIE.pdf
TIPS_NLP_Geocoding.pdf
TIPS_NLP_Google Earth Pro.pdf
TIPS_NLP_Google Earth Pro From KML to Excel.pdf