Gismo: a Generic Information Search...
...with a Mind of its Own!

LINCS Seminar

06/10/2020

Plan

A huge thanks to...

- The LINCS in general and the sk-network crew in particular
- The Maths guys from NBLF

Efficient NLP/Data analysis has never been easier!

- Text analysis / embedding (Word2Vec, FastText, Spacy)
- Ranking algorithms (PageRank / Google)
- Clustering (Louvain / scikit-network)

Do we really need yet another tool?

What is Gismo?

Gismo IS NOT

- A better embedding than word2vec or FastText
- A better ranking than PageRank
- A better clustering than Louvain

What is Gismo?

Gismo IS

- A Generic Information Search...with a Mind of its Own!
- A combination of decent embedding, ranking, and clustering that have been designed to work well together.
- A multi-purpose modular tool for corpus analysis.
- Query-based: Gismo uses queries to focus.
- Fast and scalable.
- A Python package available on pypi (`pip install gismo`).

Motto: "G" stands for Generic

Gismo is not hard-wired for a specific use case

- There is a base workflow that can be modulated/ adapted
- Examples of uses:
  - Headlines extraction
  - Ranking with redundancy removal
  - Query expansion
  - Research fields / researchers exploration...

Plan

Gismo key concepts

Gismo is powered by three main components

Dual embedding Documents are made of words... and words are made of documents. Provides relations.

Query diffusion Something important is made of important things. Provides relevance.

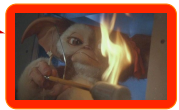
Hierarchical clustering Similar things should be merged. Recursively. Provides structure.

A document is made of words

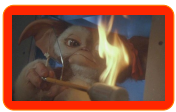
Gizmo arrives in a toy car and opens a skylight.

A document is made of words

Gizmo arrives in a toy car and opens a skylight.

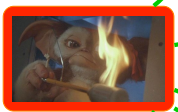


A word is made of documents



Gizmo

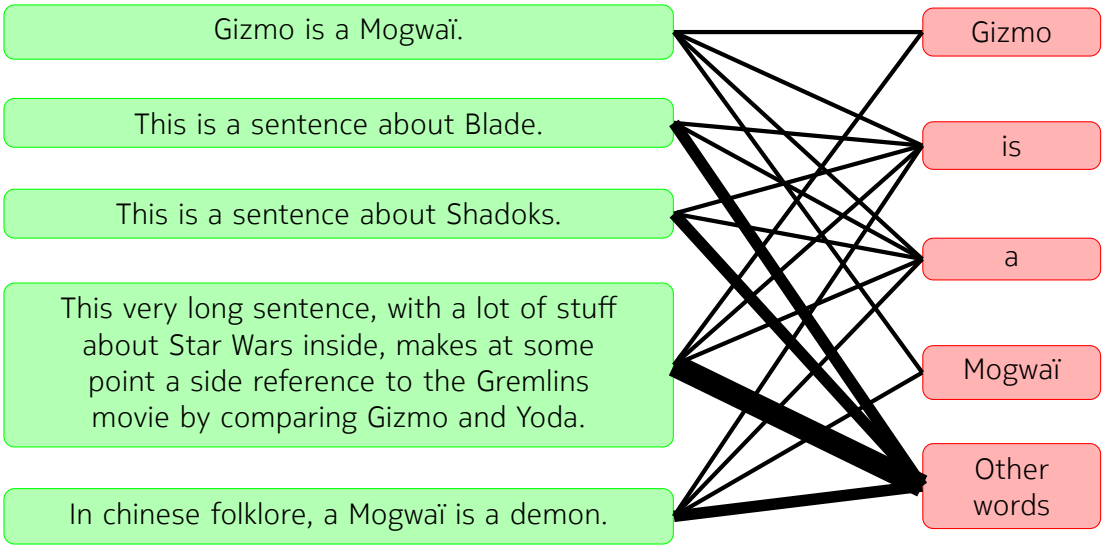
A word is made of documents



Gizmo

- Randall names the mogwai "Gizmo".
- Billys friend Pete spills a glass of water over Gizmo.
- The cocoons hatch and reptilian monsters torture Gizmo.
- Gizmo arrives in a toy car and opens a skylight.
- Mr. Wing then departs with Gizmo.

Words/documents inclusion graph



Graph weights

Edges need to be weighted

- What are the important words of a document?
- What are the important documents of a word?

TF-IDTF

- X: set of documents/paragraphs/sentences. $|X| = n$.
- Y: set of words/features/terms. $|Y| = m$.
- $G = (X \cup Y, E)$: bipartite inclusion graph; $d(.)$: degree in G.
- $x.y$: frequency of y in x (smoothed, actually).

TF-IDF (strength of y in x)

$$x.y \log\left(\frac{1+n}{1+d(y)}\right)$$

TF-ITF (strength of x in y)

$$x.y \log\left(\frac{1+m}{1+d(x)}\right)$$

TF-IDTF matrix: mutual strength between documents and words

$$w(x,y) = x.y \log\left(\frac{1+n}{1+d(y)}\right) \log\left(\frac{1+m}{1+d(x)}\right)$$

Toy example (with real values)

Words	Gismo	is	a	Mogwaï
Documents				
Gizmo is a Mogwaï.	1	1	1	1
This is a sentence about Blade.	0	1	1	0
This is a sentence about Shadoks.	0	1	1	0
This very long sentence, with a lot of stuff about Star Wars inside, makes at some point a side reference to the Gremlins movie by comparing Gizmo and Yoda.	1	0	2	0
In chinese folklore, a Mogwaï is a demon.	0	1	1	1

Toy example (with real values)

Words		Gismo	is	a	Mogwaï
Documents	ITF\IDF	0.69	0.18	0	0.69
Gizmo is a Mogwaï.	2.20	1.52	0.40	0	1.52
This is a sentence about Blade.	1.79	0	0.32	0	0
This is a sentence about Shadoks.	1.79	0	0.32	0	0
This very long sentence, with a lot of stuff about Star Wars inside, makes at some point a side reference to the Gremlins movie by comparing Gizmo and Yoda.	0.25	0.17	0	0	0
In chinese folklore, a Mogwaï is a demon.	1.63	0	0.29	0	1.14

Dual embedding

Each documents is represented by its normalized row

$$x \sim \frac{(w(x,y))_{y \in x}}{\sum_{y \in x} w(x,y)}$$

Each word is represented by its normalized column

$$y \sim \frac{(w(x,y))_{x \ni y}}{\sum_{x \ni y} w(x,y)}$$

Dual interest

- Weighted translation between X and Y enables importance diffusion.
- Space enables structure analysis on X or Y.

What should a query "Gizmo" return?

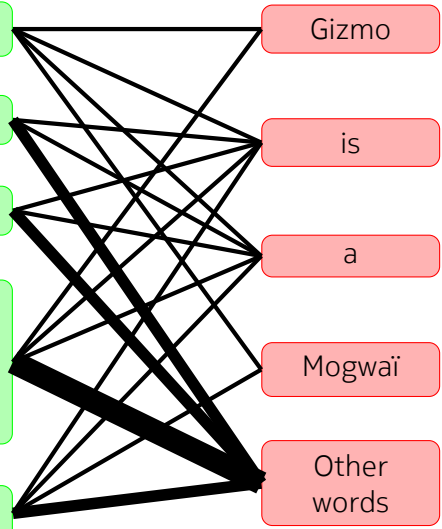
Gizmo is a Mogwaï.

This is a sentence about Blade.

This is a sentence about Shadoks.

This very long sentence, with a lot of stuff about Star Wars inside, makes at some point a side reference to the Gremlins movie by comparing Gizmo and Yoda.

In chinese folklore, a Mogwaï is a demon.



Solution: a diffusion algorithm

An old idea that works

- Propagate the query therough the graph
- Bibliometry (1950's), PageRank (1998), TextRank (2004) DIteration (2013), WalkScan (2016), Nuudle (2016)

What we do

- Mix the good ideas from related work
- Rely on the TF-IDTF space
- Provide a dual XY ranking

Gismo diffusion algorithm

Let's formalize the intuition.

Gismo diffusion algorithm

Let's formalize the intuition.

- Project query on Y (TF-IDF transformation); gives a query vector z

Gismo diffusion algorithm

Let's formalize the intuition.

- Project query on Y (TF-IDF transformation); gives a query vector Z
- Compute $R = \sum_{k \geq 1} \alpha^k Z A^k$, where A is defined on $X \cup Y$ by:
  - $a_{x,y} = \frac{\omega(x,y)}{\sum_{y \in x} \omega(x,y)}$
  - $a_{y,x} = \frac{\omega(x,y)}{\sum_{x \ni y} \omega(x,y)}$

Gismo diffusion algorithm

Let's formalize the intuition.

- Project query on Y (TF-IDF transformation); gives a query vector Z
- Compute $R = \sum_{k \geq 1} \alpha^k Z A^k$, where A is defined on $X \cup Y$ by:
  - $a_{x,y} = \frac{\omega(x,y)}{\sum_{y \in x} \omega(x,y)}$
  - $a_{y,x} = \frac{\omega(x,y)}{\sum_{x \ni y} \omega(x,y)}$
- Use R to provide a ranking on X and a ranking on Y

Gismo diffusion algorithm

Let's formalize the intuition.

- Project query on Y (TF-IDF transformation); gives a query vector z
- Compute $R = \sum_{k \geq 1} \alpha^k z A^k$, where A is defined on $X \cup Y$ by:
  - $a_{x,y} = \frac{\omega(x,y)}{\sum_{y \in x} \omega(x,y)}$
  - $a_{y,x} = \frac{\omega(x,y)}{\sum_{x \ni y} \omega(x,y)}$
- Use R to provide a ranking on X and a ranking on Y
- The rankings can also be interpreted as directions on X and Y!

Gismo diffusion algorithm

Let's formalize the intuition.

- Project query on Y (TF-IDF transformation); gives a query vector z
- Compute $R = \sum_{k \geq 1} \alpha^k z A^k$, where A is defined on $X \cup Y$ by:
  - $a_{x,y} = \frac{\omega(x,y)}{\sum_{y \in x} \omega(x,y)}$
  - $a_{y,x} = \frac{\omega(x,y)}{\sum_{x \ni y} \omega(x,y)}$
- Use R to provide a ranking on X and a ranking on Y
- The rankings can also be interpreted as directions on X and Y!
- How to choose $0 < \alpha < 1$?

Importance/relevance trade-off

Importance/relevance trade-off

- $R \sim$ a random walk from z of average length $\frac{1}{1-\alpha}$

Importance/relevance trade-off

- $R \sim$ a random walk from z of average length $\frac{1}{1-\alpha}$
- α mixes graph centrality (importance) and query proximity (relevance)

Importance/relevance trade-off

- $R \sim$ a random walk from z of average length $\frac{1}{1-\alpha}$
- α mixes graph centrality (importance) and query proximity (relevance)
- Small α: Stay close to z

Importance/relevance trade-off

- $R \sim$ a random walk from z of average length $\frac{1}{1-\alpha}$
- α mixes graph centrality (importance) and query proximity (relevance)
- Small α: Stay close to z
- Large α: explore the graph (and forgets z)

Importance/relevance trade-off

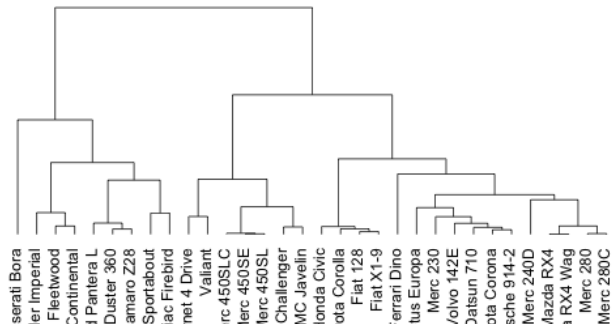
- $R \sim$ a random walk from z of average length $\frac{1}{1-\alpha}$
- α mixes graph centrality (importance) and query proximity (relevance)
- Small α: Stay close to z
- Large α: explore the graph (and forgets z)
- Current empirical choice in Gismo: 0.5 (can be adjusted)

Importance/relevance trade-off

- $R \sim$ a random walk from z of average length $\frac{1}{1-\alpha}$
- α mixes graph centrality (importance) and query proximity (relevance)
- Small α: Stay close to z
- Large α: explore the graph (and forgets z)
- Current empirical choice in Gismo: 0.5 (can be adjusted)
- Live example

Hierarchical clustering: Dendrogram

- Used in biology to study taxonomies
- Used in clustering as an alternative to k-Means
- Hot topic at LINCS
- Main issue: binary



Hierarchical clustering in Gismo: principle

Init

- k items from the same side (X or Y)
- Each item is an atomic cluster represented by a vector

Iteration

- Find clusters that are "close enough"
- Merge them

Hierarchical clustering in Gismo: secret ingredients

Query-based distortion

- Weight the TF-IDTF values with the diffusion vectors
- Similarity between documents depends on the query!

Resolution

- We use a resolution parameter to control the "close enough" condition.
- Low resolution: all nodes are merged in one single step (flat tree).
- High resolution: nodes are merged by pair (dendrogram)

See live example on Notebook

Plan

Using Gismo

- Gismo source code is on GitHub (https://github.com/baloufr/gismo)
- Documentation is on readthedocs (https://gismo.readthedocs.io/)
- Recommended install with `pip install gismo`
- Codecov: 699/700 = 99.86%
- Current version: 0.3

Performance of offline preparation

What is done there

- Use sklearn bag of words implementation
- Add IDTF weights
- Stores normalized rows and columns to speed up access

Execution time

- A few minutes for very large corpus (wikipedia, 450,000 documents, 2,000,000 words, 160,000,000 edges)
- A few seconds for large corpus (Nokia news, 11,000 documents, 22,000 words, 670,000 edges)
- Real time for small to medium live corpus

Performance of live usage

What is done there

- Project query to Y
- Diffuse between X and Y
- Extract top documents and features
- Clusterize
- Post-processing

Fast execution time

- DIteration algorithm ensures efficient computation on sparse matrix
- Numba implementation grants C-like speed

Meet the Xplorer

What is Xplorer?

- In Bell Labs, we have researchers that work on different topics
- The Xplorer is a tool powered by Gismo to navigate simultaneously researchers and topics

Data used by the Xplorer

- Dump of DBLP (titles of 5,000,000 C.S. articles, 2,500,000 unique authors).
- List of 100 researchers with DBLP id
- list of ACM domains

Plan

1. Introduction
2. Description
   - Dual Embedding
   - Query diffusion
   - Hierachical clustering
3. Gismo Python module
   - Installation
   - Performance
   - Application: the XPlorer
4. Conclusion

Conclusion

Take-Away

- Gismo: New way to explore sets of documents
- Fast and relevant analysis
- "G" stands for generic

Next steps

- Gismo 0.4 will facilitate the creation of Xplore-like tools
- Gismo is always craving for new use cases!
- Let's discuss!