Fusing Document, Collection and Label Graph-based Representations with Word Embeddings for Text Classification

Konstantinos Skianis
École Polytechnique
France

Fragkiskos D. Malliaros
CentraleSupélec & Inria Saclay
France

Michalis Vazirgiannis
École Polytechnique
France

DaSciM
Data Science and Mining Team
École Polytechnique

l'X ÉCOLE POLYTECHNIQUE UNIVERSITÉ PARIS-SACLAY

CentraleSupélec

université PARIS-SACLAY

Inria INVENTEURS DU MONDE NUMÉRIQUE

TextGraphs, NAACL-HLT, New Orleans, USA

June 8, 2018

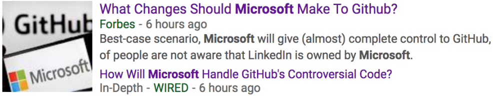
Modelling Text

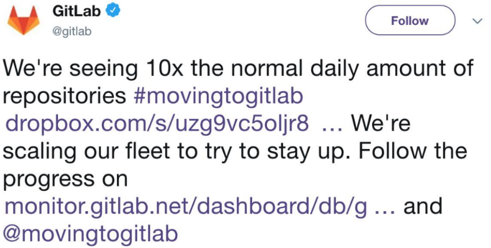
NLU Key Component

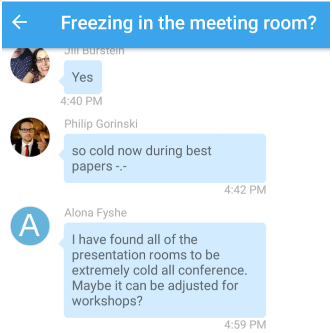
Extracting meaningful structures has always been a challenge.

We still need fast and effective ways to use text:

- real-time systems (keywords, news handling, event detection etc.)







Text Classification

Definition

Assigning categories to documents (web page, book, media articles etc.)

- TC still one of the most popular tasks (evaluation etc.)
- Spam filtering, email routing, sentiment analysis, qa, chatbots

Pipeline:

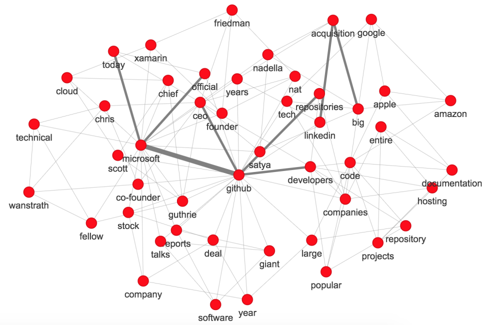
(1) Each document is modeled using the *Vector Space Model (or BoW)*

(2) Train weights regarding the importance of each term

(3) Output a class (single or multi-label, binary, multiclass)

Graph Representations

Microsoft is acquiring GitHub. After reports emerged that the software giant was in talks to acquire GitHub, Microsoft is making it official today. This is Microsoft CEO Satya Nadella's second big acquisition, following the 26.2 billion acquisition of LinkedIn two years ago. GitHub was last valued at 2 billion back in 2015, and Microsoft is paying 7.5 billion in stock for the company in a deal that should close later this year. GitHub is a large code repository that has become very popular with developers and companies hosting entire projects, documentation, and code.



https://safetyapp.shinyapps.io/GoWvis/

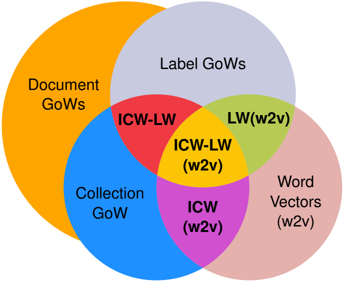
Why Graphs?

- DeepWalk (Perozzi et al., 2014)
- Graph CNNs (Duvenaud et al., 2015)
- Neural Message Passing (Gilmer et al., 2017)

Fusing Graph-of-Words with Word Embeddings

Bringing Graphs to NLP:

- Consider info about ***n***-grams
  - Expressed by paths in the graph
  - Keep the same dimensionality with BoW (compared to ***n***-grams)
- Introduce Collection-level GoW
- Blend Document, Collection and Label GoWs
- Integrate word vector similarities as weights in edges



Main Approaches

Bag-of-Words & Linear Classifiers

- Document is represented as a multiset of its terms
  ↪ fast and effective with simple classifiers
- The term independence assumption:
  ↪ disregarding co-occurence; keeping only the frequency
- ***n***-gram model (Baeza-Yates and Ribeiro-Neto, 1999)
  ↪ order of terms completely ignored, huge dimensionality

Continuous Vectors & Deep Learning

- Neural TC (Blunsom et al., 2014);(Kim, 2014)
  ↪ Current state-of-the-art results
  ↪ Large pre-trained embeddings needed
- Use the order of words with CNNs (Johnson and Zhang, 2015)
  ↪ Complex architectures with large resources (GPUs)
- Space and time limitations may arise:
  ↪ Computation can be expensive (Joulin et al., 2017)

↪ We do not focus on the classifier part, but on extracting better features.

Related Work

Popular weighting schemes:

- TF, TF-IDF (Salton and Buckley, 1988);(Singhal et al., 1996);(Robertson, 2004)
- Okapi BM25 (Robertson et al., 1995), N-gram IDF (Shirakawa et al., 2015)
- Study of frequency-based term weighting criteria (Lan et al., 2005) $\hookrightarrow$ the IDF factor is not always significant
- Delta TF-IDF for sentiment analysis (Martineau and Finin, 2009).

Bag-of-Words

Any structural information about the ordering or in general, syntactic, semantic relationship of the terms, is ignored by the weighting process.

Graph-based TC

Graph-mining for TC

- Extract frequent subgraphs (Deshpande et al., 2005);(Nikolentzos et al., 2017)
  $\hookrightarrow$ frequent subgraph mining comes with high complexity
- Random walks, other graph centrality criteria (Hassan et al., 2007);(Malliaros and Skianis, 2015)

Graph-based Text Mining, NLP and IR

- TextRank (Mihalcea and Tarau, 2004)
- Graph-of-Words (Rousseau and Vazirgiannis, 2015)
- Survey of graph-based methods in text (Blanco and Lioma, 2012)

Bag to Graph

From BoW to GoW

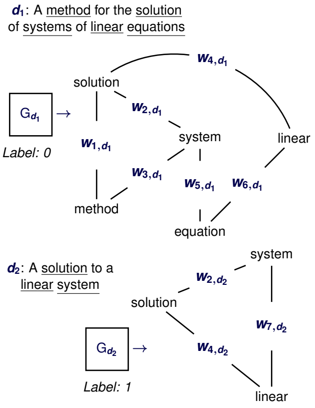
Create a graph representation for each document, where nodes represent words and edges co-occurence inside a sliding window w.

From TF-IDF to TW-ICW

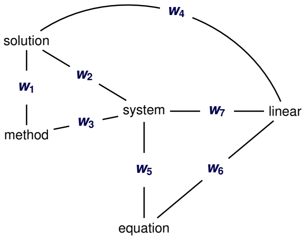
Centrality criteria

- Degree$(i) = \frac{|\mathcal{N}(i)|}{|V|-1}$.
- Closeness$(i) = \frac{|V|-1}{\sum_{j \in V} dist(i,j)}$, the sum of the length of the shortest paths between the node and all other nodes in the graph.
- Pagerank$(i) = \frac{1-\alpha}{|V|} + \alpha \sum_{\forall (j,i) \in E} \frac{\text{PR}(j)}{\text{out-deg}(j)}$

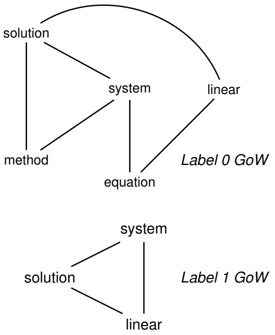
Document, Collection and Label GoWs



Document-level GoWs for d_1, d_2.



Collection-level GoW $\mathcal{G}$.



Label GoWs for two classes.

Proposed Weighting Schemes

Having the collection GoW, we derive the "Inverse Collection Weight" metric:

$$\text{ICW}(\boldsymbol{t}, \mathcal{D}) = \frac{\mathbf{max}_{v \in \mathcal{D}} \text{TW}(\boldsymbol{v}, \mathcal{D})}{\text{TW}(\boldsymbol{t}, \mathcal{D})}$$

Then, the TW-ICW metric becomes:

$$\text{TW-ICW}(\boldsymbol{t}, \boldsymbol{d}) = \text{TW}(\boldsymbol{t}, \boldsymbol{d}) \times \mathbf{log}(\text{ICW}(\boldsymbol{t}, \mathcal{D}))$$

For labels, our weighting scheme is a variant of TW-CRC:

$$\text{LW}(\boldsymbol{t}) = \frac{\mathbf{max}(\text{deg}(\boldsymbol{t}, \boldsymbol{L}))}{\mathbf{max}(\text{avg}(\text{deg}(\boldsymbol{t}, \boldsymbol{L})), \mathbf{min}(\text{deg}(\boldsymbol{L})))}$$

Last, the TW-ICW-LW metric becomes:

$$\text{TW-ICW-LW}(\boldsymbol{t}, \boldsymbol{d}) = \text{TW}(\boldsymbol{t}, \boldsymbol{d}) \times \mathbf{log}(\text{ICW}(\boldsymbol{t}, \mathcal{D}) \times \text{LW}(\boldsymbol{t}))$$

Edge Weighting using Word Embeddings

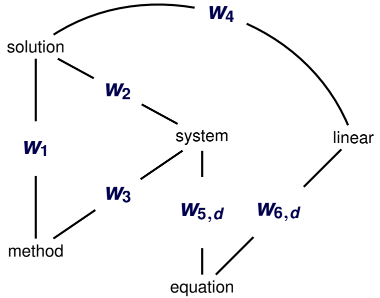
Taking the most-out-of graphs via word vectors

Use rich word embeddings in order to extract relationships between terms.

- Inject similarities as weights on edges
  - Reward semantically close words in the document GoW (TW)
  - Penalize them in the collection GoW (ICW)

$$w(t_1, t_2) = 1 - \frac{\text{sim}^{-1}(t_1, t_2)}{\pi}$$

d_1: A method for the solution of systems of linear equations



Datasets & Set-up

- Linear SVMs with grid search cross-validation for tuning the ***C*** parameter.
- Removed stopwords.
- No stemming or lowercase transformation, to match `word2vec`.
- Multi-core document and collection graph construction.

	Train	Test	Voc	Avg	#w2v	#ICW
IMDB	1,340	660	32,844	343	27,462	352K
WebKB	2,803	1,396	23,206	179	20,990	273K
20NG	11,293	7,528	62,752	155	54,892	1.7M
Amazon	5,359	2,640	19,980	65	19,646	274K
Reuters	5,485	2,189	11,965	66	9,218	163K
Subj.	6,694	3,293	8,639	11	8,097	58K

#ICW: number of edges in the collection-level graph; #w2v: number of words in pre-trained vectors.

Results

Macro-F1 and accuracy for window size w. Bold for best performance on each window size and blue for best overall on a dataset. * indicates stat. significance of improvement over TF at $p < 0.05$ using micro sign test.

Methods	20NG (MAX)				IMDB (SUM)				SUBJECTIVITY (MAX)			
	$w = 3$		$w = 4$		$w = 2$		$w = 3$		$w = 6$		$w = 7$	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
TF	80.88	81.55	-		84.23	84.24	-		88.42	88.43	-	
w2v	74.43	75.75	-		82.57	82.57	-		87.67	87.67	-	
TF-binary (ngrams)	81.64	82.11*	-		83.02	83.03	-		87.51	87.51	-	
TW (degree)	82.37	83.00*	82.21	82.83*	84.82	84.84	84.67	84.69	88.33	88.33	89.00	89.00*
TW (w2v)	81.88	82.51*	82.21	82.87*	84.66	84.69	84.52	84.54	87.75	87.57	87.66	87.67
TF-IDF	82.44	83.01*	-		83.33	83.33	-		89.06	89.06*	-	
TF-IDF-w2v	82.52	83.09*	-		82.87	82.87	-		89.91	89.91*	-	
TW-IDF (degree)	84.75	85.47*	84.80	85.46*	82.86	82.87	83.02	83.03	89.33	89.34*	89.33	89.34*
TW-IDF (w2v)	84.66	85.32	84.46	85.13	83.47	83.48	83.31	83.33	86.42	86.42	86.51	86.51
TW-ICW (deg, deg)	85.24	85.80*	**85.41**	**86.05***	84.98	85.00	85.13	85.15	89.30	89.31*	89.61	89.61*
TW-ICW (w2v)	**85.33**	**85.93***	85.29	85.90*	85.12	85.15	84.82	84.84	89.61	89.61*	87.30	87.30
TW-ICW-LW (deg)	85.01	85.66*	85.02	85.66*	85.73	85.75	85.28	85.30	**90.12**	**90.13***	**90.27**	**90.28***
TW-ICW-LW (w2v)	82.56	83.11*	82.24	82.81*	85.29	85.30	84.39	84.39	87.70	87.70	87.70	87.70
TW-ICW-LW (pgr)	83.92	84.66	83.80	84.54	84.97	85.00	85.73	85.75	86.60	86.60	86.45	86.45
TW-ICW-LW (cl)	84.61	85.22	84.71	85.27	**87.27**	**87.27***	**86.06**	**86.06**	89.97	89.97*	90.09	90.10*

Results (2/2)

Macro-F1 and accuracy for window size w. Bold for best performance on each window size and blue for best overall on a dataset. * indicates stat. significance of improvement over TF at $p < \mathbf{0.05}$ using micro sign test.

Methods	AMAZON (MAX)				WEBKB (SUM)				REUTERS (MAX)			
	$w = 2$		$w = 3$		$w = 2$		$w = 3$		$w = 2$		$w = 3$	
	F1	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**
TF	80.68	80.68	-		90.31	91.91	-		91.51	96.34	-	
w2v	79.05	79.05	-		84.54	86.58	-		91.35	96.84	-	
TF-binary (ngrams)	79.84	79.84	-		91.22	92.85	-		86.33	95.34	-	
TW (degree)	80.07	80.07			91.69	92.64	91.45	92.49	93.58	97.53*	93.08	97.25*
TW (w2v)	80.07	80.07	79.54	79.54	91.70	92.64	91.00	92.06	93.09	97.35*	93.43	97.25*
TF-IDF	80.26	80.26	-		87.79	89.89	-		91.89	96.71	-	
TF-IDF-w2v	80.49	80.49	-		88.18	90.18	-		91.33	96.80	-	
TW-IDF (degree)	81.47	81.47*	81.55	81.55*	90.38	91.70	90.47	91.84	93.80	97.30*	93.13	97.35*
TW-IDF (w2v)	79.61	79.62	77.60	77.61	90.81	92.20	90.60	91.91	93.38	97.44*	**93.87**	**97.44***
TW-ICW (deg, deg)	82.08	82.08*	82.02	82.02*	91.72	92.78	91.42	92.49	92.91	97.35	93.59	97.39*
TW-ICW (w2v)	80.86	80.87*	78.82	78.82	91.58	92.64	91.84	92.85	93.57	97.30*	92.96	97.25
TW-ICW-LW (deg)	82.72	82.72*	82.91	82.91*	91.86	92.92	91.95	92.92	**93.88**	**97.53***	93.48	97.35*
TW-ICW-LW (w2v)	80.56	80.56	78.32	78.33	90.74	91.99	90.01	91.34	92.51	96.89	92.14	96.98
TW-ICW-LW (pgr)	82.23	82.23*	82.46	82.46*	91.18	92.20	92.23	93.07	93.38	97.35*	93.37	97.35*
TW-ICW-LW (cl)	**82.90**	**82.91***	**83.02**	**83.03***	**92.72**	**93.57***	**92.86**	**93.57***	93.12	97.25	92.87	97.21

Comparison vs state-of-the-art methods

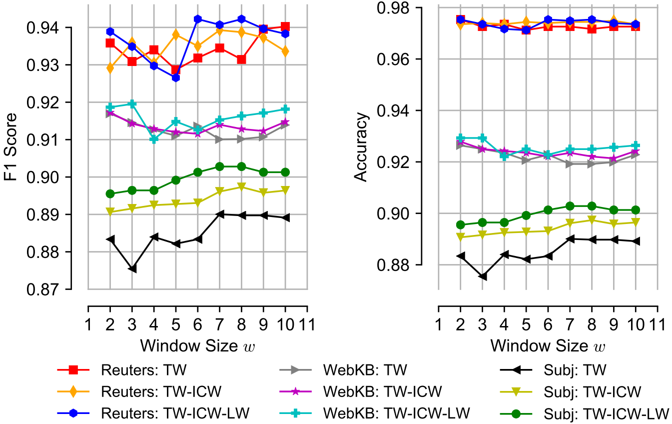
	20NG	IMDB	SUBJ.	AMAZON	WEBKB	REUTERS
CNN (no w2v, 20 ep.) (Kim, 2014)	83.19	74.09	88.16	80.68	88.17	94.75
FastText (100 ep.) (Joulin et al., 2017)	79.70	84.70	88.60	79.50	92.60	97.00
TextRank (Mihalcea and Tarau, 2004)	82.56	83.33	84.78	80.49	92.27	97.35
Word Attraction (Wang et al., 2015)	61.24	70.75	86.60	78.29	79.46	91.34
TW-CRC (Shanavas et al., 2016)	85.35	85.15	89.28	81.13	92.71	97.39
TW-ICW-LW (**ours**)	**86.05**	**87.27**	**90.28**	**83.03**	**93.57**	**97.53**

Comparison in accuracy(%) to deep learning and graph-based approaches.

Notes

- CNN with non-static random embeddings, multichannel.
- Optimal settings not searched.
- Early stopping, or multiple architectures proposed.

Examining Window Size



F1 score (left) and accuracy (right) of TW, TW-ICW and TW-ICW-LW (all degree) on REUTERS, WEBKB and SUBJECTIVITY, for $\boldsymbol{w = \{2, \ldots, 10\}}$.

Discussion

- TW-ICW-LW: best in 5/6 datasets.
- TW-ICW and TW-ICW-LW: Best in 6/6
- When label graphs are used, *word2vec* does not improve the accuracy.
  $\hookrightarrow$ terms concerning different labels can be close in the word vector space.
- Closeness in document GoW $\longrightarrow$ best performance in 3/6.
  $\hookrightarrow$ can only have an affect in larger document lengths and when used along with label graphs.

Conclusion

Contribution

- A full graph-based framework for TC
- Determine the importance of a term using node centrality criteria
  - Document, collection and label level schemes, that penalize globally important terms and reward locally important terms respectively
- Incorporate additional word-embedding information as weights in the graph-based representations

Future Directions

- Sentence, Paragraph, Topic GoWs
- Could also be applied in IR(keyword extraction), summarization etc.
  - Other centralities may affect tasks differently
- Unsupervised: community detection algorithms to identify clusters of words or documents in collection GoW
- *Graph-of-Documents*
  - Graph comparison via graph kernels (Borgwardt et al., 2007)
  - Word Mover's Distance (Kusner et al., 2015)
- Graph-based representations of text could also be fitted into deep learning architectures (Lei et al., 2015).
- Neural Message Passing (Gilmer et al., 2017)
- Word embeddings:
  - Topical Word Embeddings (Liu et al., 2015)
  - ELMo (Peters et al., 2018)

Thank you!

Code: github.com/y3nk0/Graph-Based-TC

Konstantinos Skianis
Data Science and Mining Group (DaSciM)
École Polytechnique, France
kskianis@lix.polytechnique.fr
www.lix.polytechnique.fr/∼kskianis

Fragkiskos D. Malliaros
CentraleSupélec & Inria Saclay, France
fragkiskos.malliaros@centralesupelec.fr
http://fragkiskos.me/

Michalis Vazirgiannis
Data Science and Mining Group (DaSciM)
École Polytechnique, France
mvazirg@lix.polytechnique.fr
www.lix.polytechnique.fr/∼mvazirg