Sentiment Analysis (**PML** Ch. 8) (and Recent Advances in Textual Analysis) Machine Learning for Finance (FIN 570)

Instructor: Jaehyuk Choi

Peking University HSBC Business School, Shenzhen, China

2023-24 Module 3 (Spring 2024)

Introduction

Background

- Text corpus (a large set of documents) is an attractive data source for research.
- Text data may provide novel measures to address research questions.

Takeaway

- Word embedding methods (e.g., `Word2Vec`, `GloVe` or `BERT`) based on machine learning has recently been developed and used in research.
- With the codes and pre-trained models (e.g., `Gensim`) available, using those methods for research is much easier than you think!

Disclaimer

- This talk focuses on methods rather than research questions.
- Examples are biased to finance.

Traditional dictionary approach

- Want to score the emotion or opinion in text for a specific purpose.
- The score based on simple word counting:

$$\text{Score} = \frac{\#\text{ of positive words} - \#\text{ of negative words}}{\#\text{ of positive words} + \#\text{ of negative words}},$$

  where **positive** and **negative** words are predefined for the purpose of the problem.
- LM dictionary (Loughran and Mcdonald, 2011, 2016) is widely used in finance research (Data Source)
- Limitation: very unlikely that such 'classification dictionary' exists for your own question. E.g., Hawkish v.s. Dovish monetary policy.

Word Embedding

- Want to map the meaning of text data (word/sentence/document) into a numerical vector (embedding).
- Ideally, the words with similar meanings should be located near in the vector space.
- The similarity between two words is typically measured by cosine similarity:

$$\text{Sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}|\,|\mathbf{y}|} = \cos\theta.$$

- The vectors are still raw data. They are used as an input to further analysis (e.g., classification dictionary, score, regression, ML, etc.).

Bag-of-words: one-hot-encoding with n-gram

- Dummy variables for all unique words in the dictionary.
- The vector dimension is as big as the number of words.
- Vector $\mathbf{v}_i$ of sentence/document has component value 1 if a word is used or 0 otherwise.
- Order of words (syntactic relation) is ignored.
  "Jane likes John" v.s. "John likes Jane" treated same.
- Meaning of words (semantic relation) is ignored.
  Zero cosine similarity for any two different words.
- The **words** can be generalized to the sequence of multiple words.
  - **1-gram model**: one word. E.g., "the", "sun", "is", "shining"
  - **2-gram model**: two words. E.g., "the sun", "sun is", "is shining"
- Still an effective approach for a simple task: e.g., SPAM filtering

Pre-processing (a.k.a., tokenizing)

- **Stemming**: E.g., run, runs, ran $\Rightarrow$ run
- **Removing stop words**: Remove e.g. "is", "and", "had", and "like".
- Tricky word segmentation in Chinese: Jieba (Python).

Bag-of-words: TF-IDF

A weighting scheme to improve one-hot-encoding:

- The words frequently repeated in a document are important.
- The words common across documents are less relevant.

Term frequency: $tf(t, d)$

$$tf(t, d) = \text{the frequency of a term } t \text{ appearing in document } d.$$

Inverse document frequency: $idf(t, D)$

When $df(t)$ is **document frequency**, # of docs containing the term t.

$$idf(t) = \log \frac{n_D}{1 + df(t)} \quad \text{or} \quad \log \frac{1 + n_D}{1 + df(t)},$$

Term frequency-inverse document frequency (tf-idf)

$$tf\text{-}idf(t, d) = tf(t, d) \times idf(d) \quad \text{or} \quad tf(t, d) \times \big(1 + idf(d)\big)$$

Example: source

1. "The sky is blue." $\Rightarrow$ sky, blue
2. "The sun is bright today." $\Rightarrow$ sun, bright, today
3. "The sun in the sky is bright." $\Rightarrow$ sun, sky, bright
4. "We can see the shining sun, the bright sun." $\Rightarrow$ can, see, shining, sun, bright, sun

$\texttt{tf}(t, d)$

	blue	bright	can	see	shining	sky	sun	today
1	1/2	0	0	0	0	1/2	0	0
2	0	1/3	0	0	0	0	1/3	1/3
3	0	1/3	0	0	0	1/3	1/3	0
4	0	1/6	1/6	1/6	1/6	0	1/3	0

x

$\texttt{idf}(t, D)$

	blue	bright	can	see	shining	sky	sun	today
	0.602	0.125	0.602	0.602	0.602	0.301	0.125	0.602

- TF-IDF: Multiply TF and IDF scores, use to rank importance of words within documents
  - Most important word for each document is highlighted

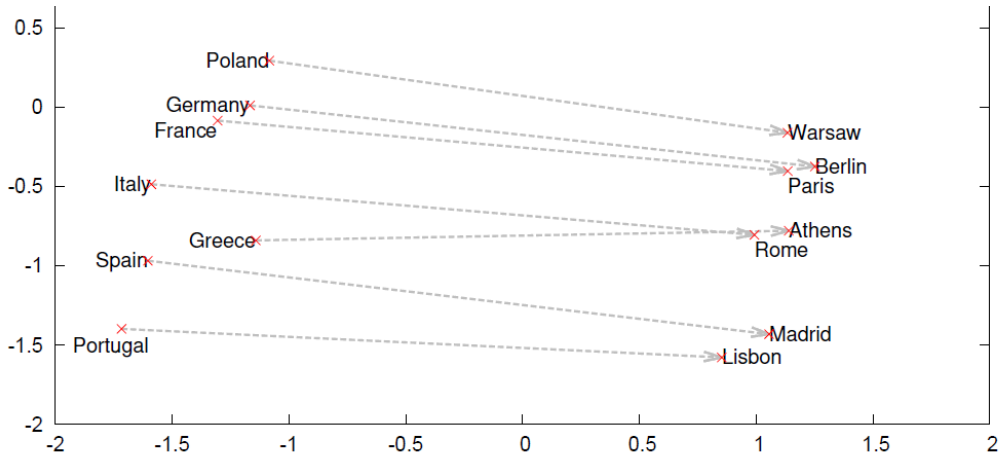
$\texttt{tfidf}(t, d, D) = \texttt{tf}(t, d) \cdot \texttt{idf}(t, D)$

	blue	bright	can	see	shining	sky	sun	today
1	**0.301**	0	0	0	0	0.151	0	0
2	0	0.0417	0	0	0	0	0.0417	**0.201**
3	0	0.0417	0	0	0	**0.100**	0.0417	0
4	0	0.0209	**0.100**	**0.100**	**0.100**	0	0.0417	0

Word2Vec

Mikolov, Chen, Corrado, and Dean (2013a, Google) Efficient Estimation of Word Representations in Vector Space. *arXiv*.

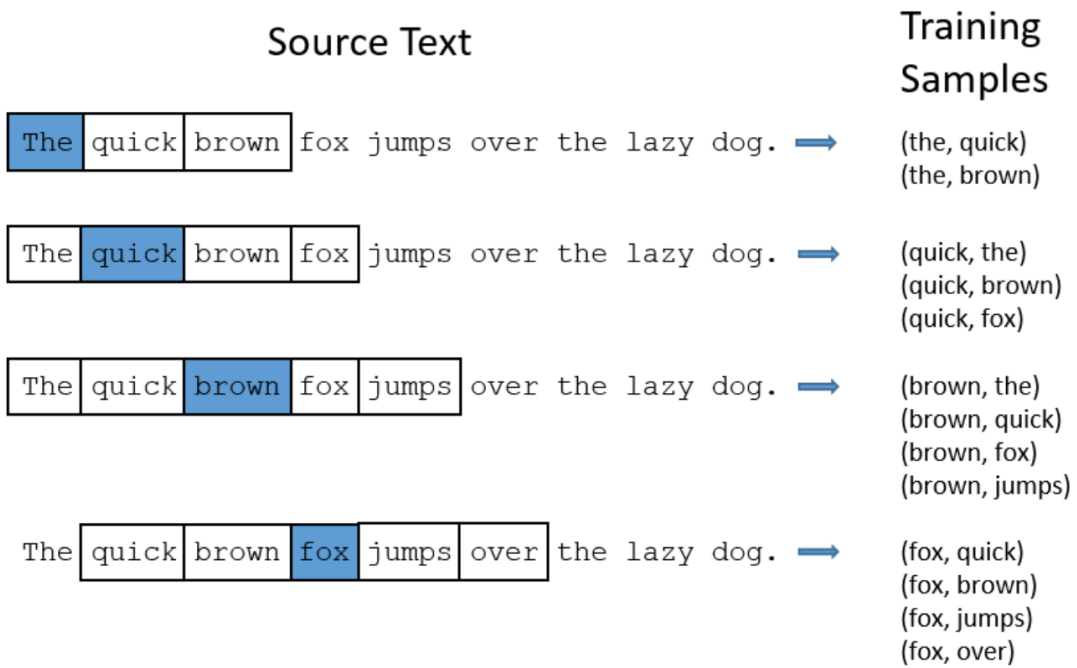
- A breakthrough in word embedding algorithm.
- The vector dimension is typically 200∼500, much smaller than BoW.
- Generates word vectors that preserve the relationship between words.
  - V('King') − V('Man') + V('Woman') ≈ ???
  - V('Madrid') − V('Spain') + V('France') ≈ ???



Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Source: Mikolov et al. (2013b)

- Co-occurrence: the words appearing in similar neighbors are likely to have a similar meaning.
- A (shallow) neural-network-based unsupervised learning to learn the relationship between words.
- Vectors are iteratively adjusted to maximize the likelihood of predicting the center word with the surrounding words through the corpus (CBOW).
- Extended to sentences and documents: `Doc2Vec` (Le and Mikolov, 2014)



Limitation of Word2Vec

- Word order not considered.
- Assumes each word has a unique meaning (1-to-1 mapping to vector). "Go to bank to withdraw cash" v.s. "Go to river bank"

GloVe (Global Vectors)

Pennington, Socher, and Manning (2014, Stanford) GloVe: Global vectors for word representation. In: the 2014 EMNLP Proceedings. pp 1532–1543

- Different algorithm, but basic idea and performance are comparable to Word2Vec.
- Various pre-trained models available in Gensim (List).
  - glove-twitter-XXX
  - glove-wiki-gigaword-XXX
  - v.s. word2vec-google-news-300

BERT

Devlin, Chang, Lee, and Toutanova (2019, Google) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv*.

- Bidirectional Encoder Representations from Transformers.
- Currently used in all English searches in Google.
- v.s. ELMo (Embeddings from Language Models).

Source: Web

- Dominant performance over other methods including `Word2Vec/GloVe`.
- Fully exploits deep-learning architecture.
- Context-aware embedding (v.s. context-independent embedding in `Word2Vec/GloVe`)
- Two-stage: Pre-training and Fine-tuning
  - **Masked Language Model**: Predict 15% of masked words from context
  - **Next Sentence Prediction**: Given two sentences, tell if they are consecutive.

Case 1: H-P product similarity (BoW: One-hot-encoding)

Hoberg and Phillips (2016). Text-Based Network Industries and Endogenous Product Differentiation. *JPE*, 124(5), 1423–1465.

- Analyze firms' business description sections from SEC 10-K filing.
- Construct doc vectors with one-hot-encoding bag-of-words model:
  - Create a dictionary of nouns and proper nouns used in all 10-K.
  - Omit the common words used in 20% or more firms (c.f., tf-idf).
- Calculate the product (cosine) similarity for firm pairs.
- Define the text-based network industry classifications (TNIC). TNIC works better than classical industry classification (NAICS or SIC).
- Similarity data (in different levels) available in H-P Data Library
- Sehrawat (2019) Learning Word Embeddings from 10-K Filings for Financial NLP Tasks. *WP*. Pre-trained `Word2Vec` data available in Github (PyTorch).

Case 2: Measuring corporate culture with `Word2Vec`

Li, Mai, Shen, and Yan (2021b). Measuring Corporate Culture Using Machine Learning. *RFS*, 34(7), 3265–3315.

See also Li, Liu, Mai, and Zhang (2021a) The Role of Corporate Culture in Bad Times: Evidence from the COVID-19 Pandemic. *JFQA* 56:2545–2583.

- Use `Word2Vec` method to analyze the firm's earnings call transcripts.
- Measures corporate culture in five values:
  innovation, integrity, quality, respect, and teamwork.
  - Begin with the seed words of each value. E.g., **teamwork**: collaborate, collaboration, collaborative, cooperate, cooperation, cooperative, and teamwork. Calculate the average of the word vectors of the seed words.
  - Find the top 500 words ('culture dictionary') with the highest similarity to the average.
  - Measure the corporate culture value as the sum of $tf\text{-}idf(d,t)$ for the words t in the each culture dictionary.

Table 2
Thirty most representative and most frequently occurring words in the culture dictionary

A. Thirty most representative words for each cultural value in the culture dictionary

Innovation	Integrity	Quality	Respect	Teamwork
Creativity	Accountability	Dedicated	Talented	Collaborate
Innovative	Ethic	Quality	Talent	Cooperation
Innovate	Integrity	Dedication	Empower	Collaboration
Innovation	Responsibility	Customer_service	Team_member	Collaborative
Creative	Transparency	Customer	Employee	Cooperative
Excellence	Accountable	Dedicate	Team	Partnership
Passion	Governance	Service_level	Leadership	Cooperate
World-class	Ethical	Mission	Leadership_team	Collaboratively
Technology	Transparent	Service_delivery	Culture	Partner
Operational_excellence	Trust	Customer_satisfaction	Teammate	Co-operation
Passionate	Responsible	Service	Organization	Coordination
Product_innovation	Oversight	Reliability	Entrepreneurial	Engage
Capability	Independence	Commitment	Skill	Jointly
Customer_experience	Objectivity	Customer_need	Executive	Coordinate
Thought_leadership	Moral	Customer_support	Empowerment	Teamwork
Expertise	Trustworthy	High-quality	Management_team	Business_partner
Agility	Fairness	Ensure	Best_brightest	Alliance
Efficient	Hold_accountable	Customer_relationship	Professionalism	Team_up
Technology_innovation	Corporate_governance	Quality_service	Staff	Technology_partner
Competency	Autonomy	Product_quality	Highly_skilled	Joint
Know-how	Core_value	Quality_product	Skill_set	Cooperatively
Cutting-edge	Assure	Capable	Technologist	Relationship
Agile	Stakeholder	Service_quality	Competent	Collaborator
Creatively	Fiduciary_responsibility	End_user	Entrepreneur	Interaction
Customer-centric	Continuity	Quality_level	Experienced	Working_relationship
Enable	Credibility	Customer_expectation	Energize	Co-operate
Value_proposition	Honesty	Service_capability	Entrepreneurial_spirit	Technology_partnership
Reinvent	Privacy	Client	High-caliber	Association
Focus	Fiduciary_duty	Customer_requirement	Manager	Dialogue
Innovation_capability	Rigor	Sla	Leadership_skill	Dialog

Source: Li et al. (2021b)

Case 3: Patent similarity with Doc2Vec

Whalen, Lungeanu, DeChurch, and Contractor (2020). Patent Similarity Data and Innovation Metrics. *J. of Empirical Legal Studies* 17:615–639.

- Train Doc2Vec (with Gensim) on the US utility patents (USPTO)
- 300-dimensional vector representations of extensive patents.
- Patent vectors and 100 most-similar patents for each patent available in Github.

Work-in-progress by Celil, Choi & Selvam

- From Whalen's patent vectors, generate a firm-level innovation vector.
- Construct (time series of) innovation similarity of firm pairs.
- Find that the funds holding portfolio concentrated in innovation space outperform (and more).

Case 4: BERT fine-tuned to finance corpus

Huang, Wang, and Yang (2020, HKUST) FinBERT—A Deep Learning Approach to Extracting Textual Information. *WP*.

- "incorporates the contextual relations between words in the finance domain."
- "Fine-tuned" to 10,000 analyst report sentences previously classified by researchers (Huang et al., 2014).
- Pre-trained model (and sentiment calculation code) available in Github.
- "FinBERT significantly outperforms the LM dictionary, the naive Bayes, and Word2Vec" in understanding analyst reports and earnings calls.

Conclusion

Takeaway

- Word embedding methods (e.g., `Word2Vec`, `GloVe` or `BERT`) based on machine learning has recently been developed and used in research.
- With the codes and pre-trained models (e.g., `Gensim`) available, using those methods for research is much easier than you think!

Thank you for
your attention!

References I

Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:181004805 [cs] URL http://arxiv.org/abs/1810.04805, 1810.04805

Hoberg G, Phillips G (2016) Text-Based Network Industries and Endogenous Product Differentiation. Journal of Political Economy 124(5):1423–1465, doi:10.1086/688176

Huang A, Wang H, Yang Y (2020) FinBERT—A Deep Learning Approach to Extracting Textual Information. SSRN Scholarly Paper ID 3910214, Social Science Research Network, Rochester, NY, doi:10.2139/ssrn.3910214

Huang AH, Zang AY, Zheng R (2014) Evidence on the Information Content of Text in Analyst Reports. The Accounting Review 89(6):2151–2180, doi:10.2308/accr-50833

Le Q, Mikolov T (2014) Distributed Representations of Sentences and Documents. In: International Conference on Machine Learning, pp 1188–1196, URL https://proceedings.mlr.press/v32/le14.html

Li K, Liu X, Mai F, Zhang T (2021a) The Role of Corporate Culture in Bad Times: Evidence from the COVID-19 Pandemic. Journal of Financial and Quantitative Analysis 56(7):2545–2583, doi:10.1017/S0022109021000326

Li K, Mai F, Shen R, Yan X (2021b) Measuring Corporate Culture Using Machine Learning. The Review of Financial Studies 34(7):3265–3315, doi:10.1093/rfs/hhaa079

References II

Loughran T, Mcdonald B (2011) When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. The Journal of Finance 66(1):35–65, doi:10.1111/j.1540-6261.2010.01625.x

Loughran T, Mcdonald B (2016) Textual Analysis in Accounting and Finance: A Survey. Journal of Accounting Research 54(4):1187–1230, doi:10.1111/1475-679X.12123

Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient Estimation of Word Representations in Vector Space. arXiv:13013781 [cs] URL http://arxiv.org/abs/1301.3781, 1301.3781

Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013b) Distributed Representations of Words and Phrases and their Compositionality. arXiv:13104546 [cs, stat] URL http://arxiv.org/abs/1310.4546, 1310.4546

Pennington J, Socher R, Manning CD (2014) GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp 1532–1543

Sehrawat S (2019) Learning Word Embeddings from 10-K Filings for Financial NLP Tasks. SSRN Scholarly Paper ID 3480902, Social Science Research Network, Rochester, NY, doi:10.2139/ssrn.3480902

Whalen R, Lungeanu A, DeChurch L, Contractor N (2020) Patent Similarity Data and Innovation Metrics. Journal of Empirical Legal Studies 17(3):615–639, doi:10.1111/jels.12261