Deep Natural Language Processing for Search Systems




Weiwei Guo
Huiji Gao
Jun Shi
Bo Long

Agenda

1. Introduction
2. Deep Learning for Natural Language Processing
3. Deep NLP in Search Systems
4. Real World Examples

Agenda

1. **Introduction**
2. Deep Learning for Natural Language Processing
3. Deep NLP in Search Systems
4. Real World Examples

Introduction

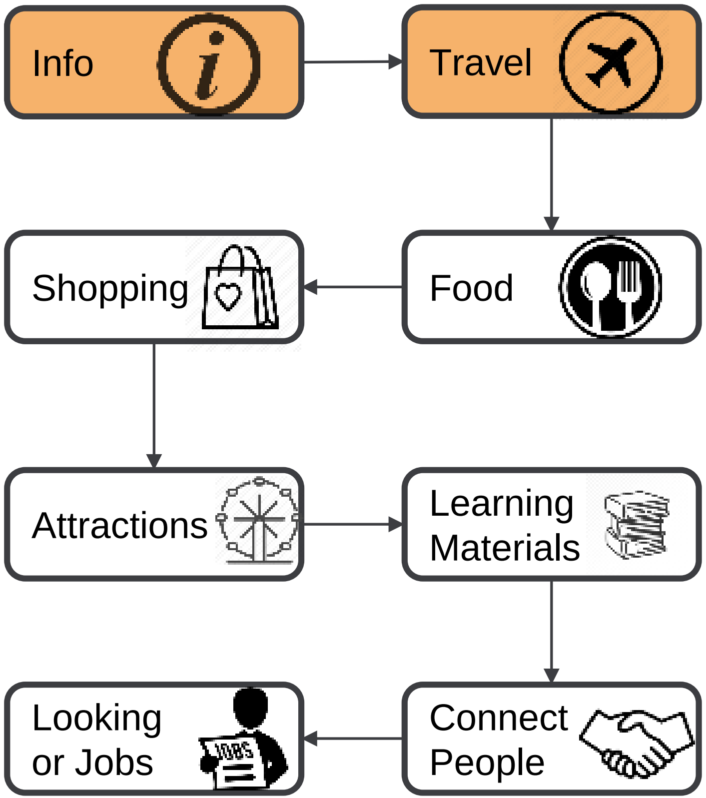
Huiji Gao

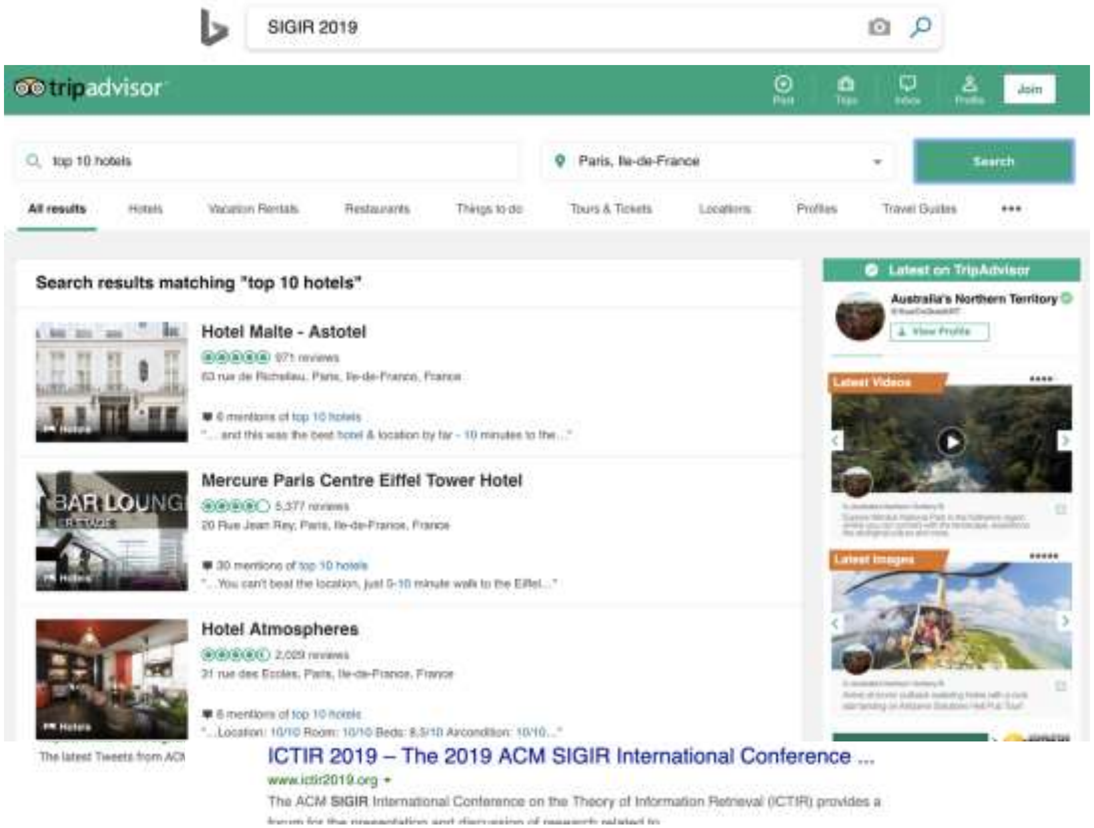
Introduction - Search Systems

Search is Everywhere

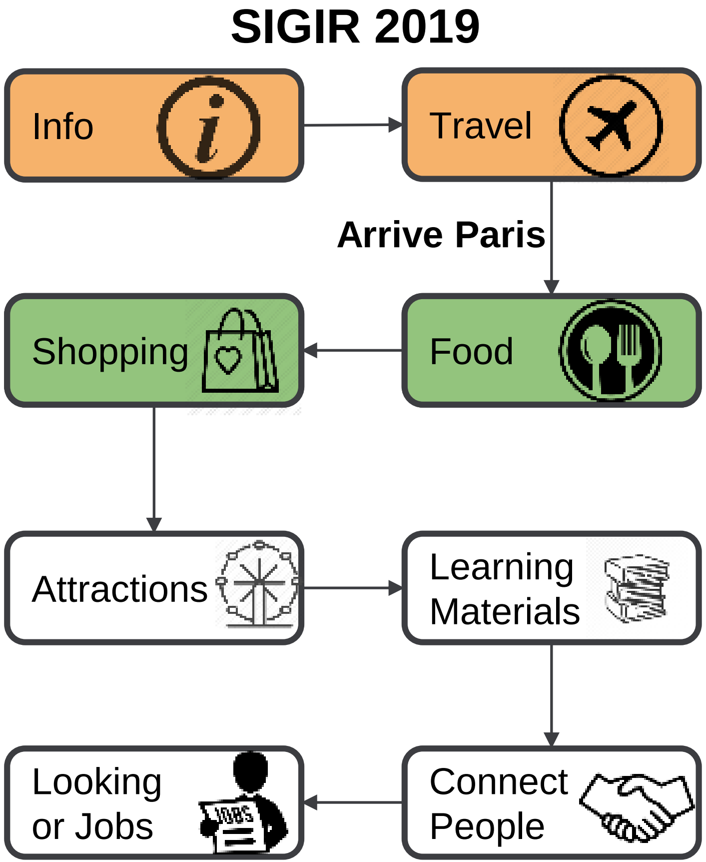
Introduction - Search Systems

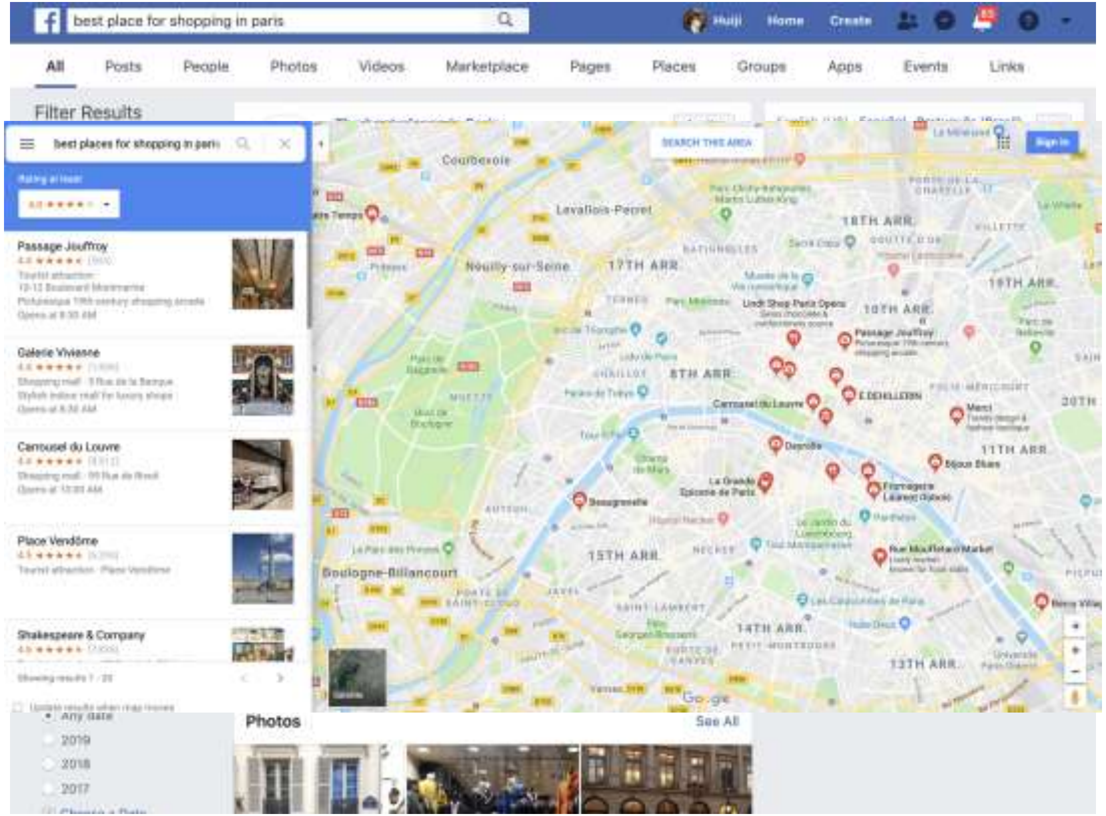
SIGIR 2019



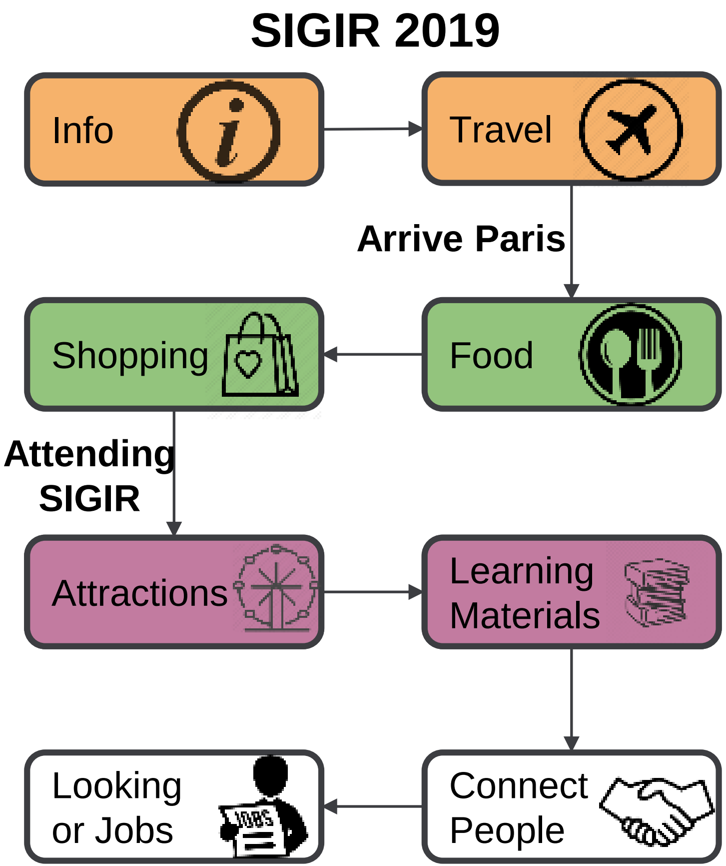


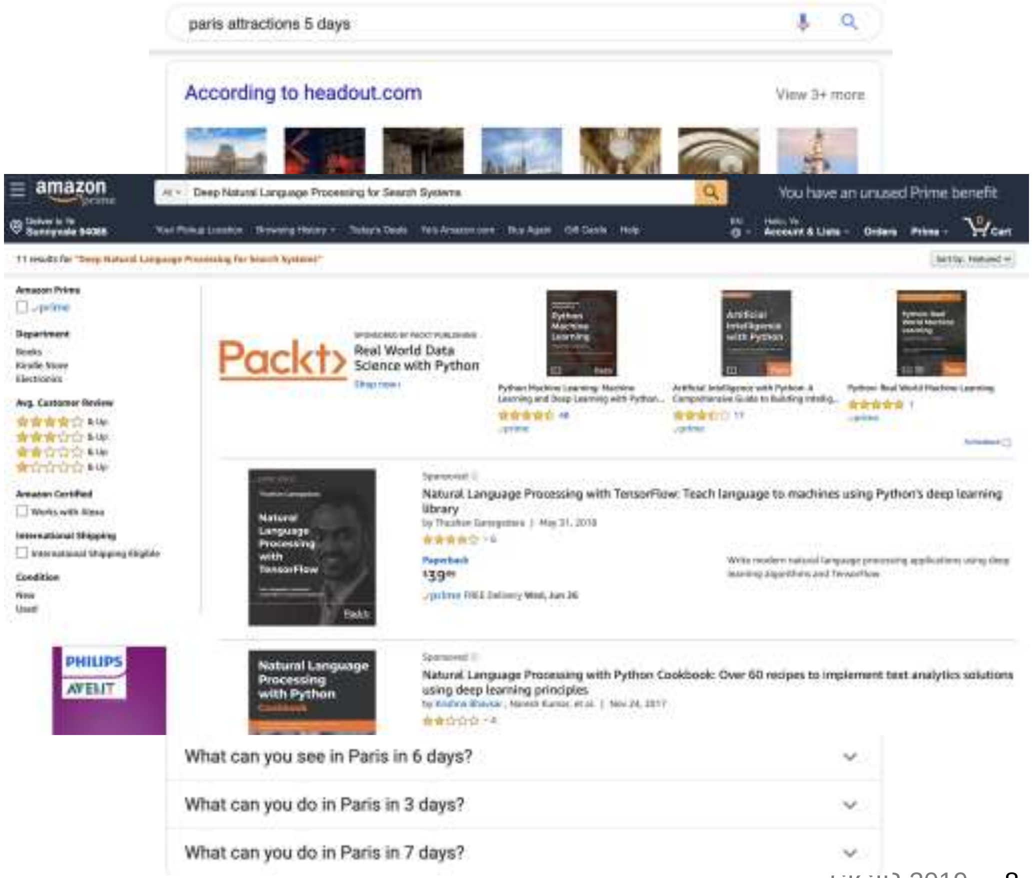
Introduction - Search Systems





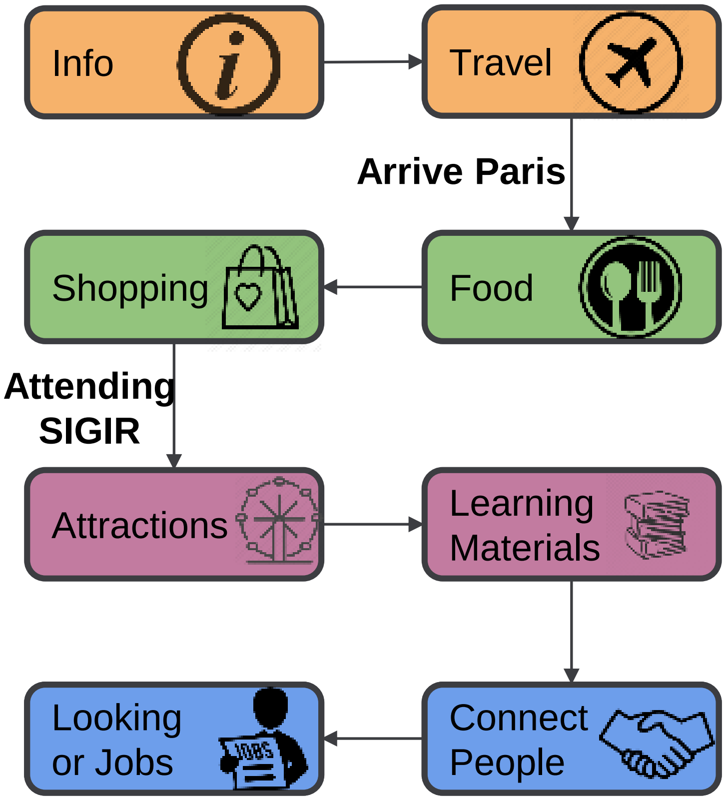
Introduction - Search Systems

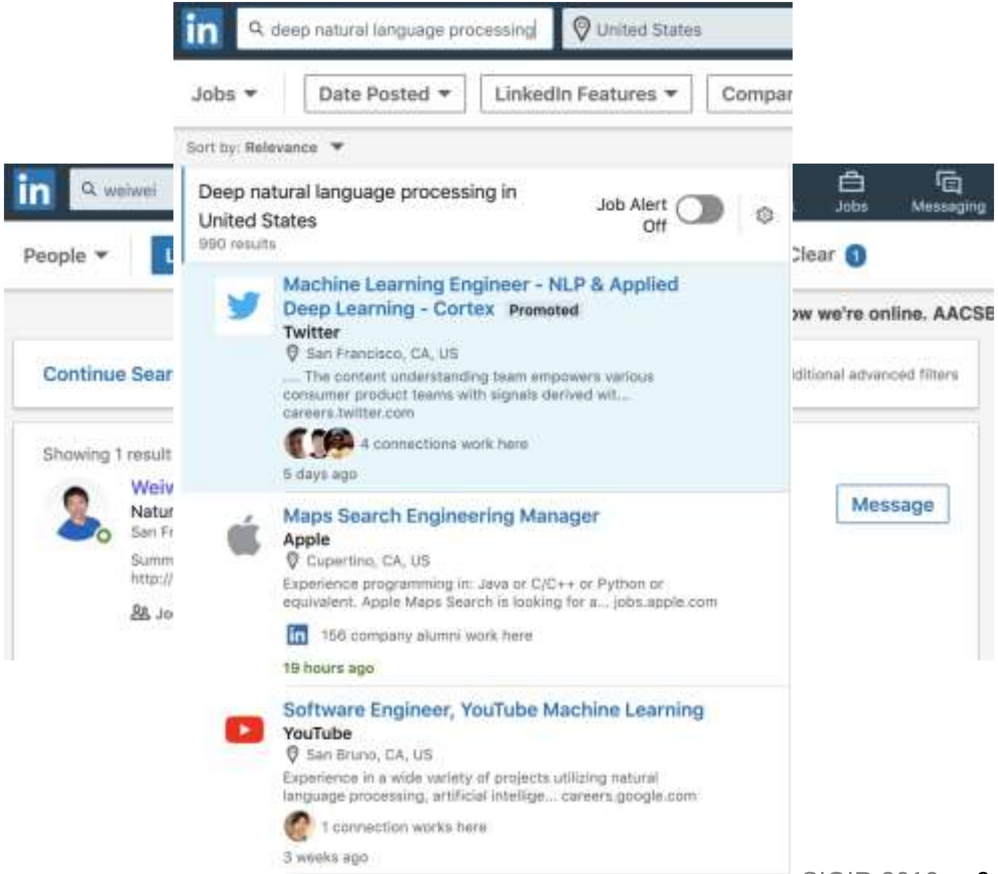




Introduction - Search Systems

SIGIR 2019





Introduction - NLP in Search Systems

- **Understand Searcher Intention**
  - Search Queries
  - User Profiles

Best Places to Shop in Paris

Budget? Gender? Location? ...

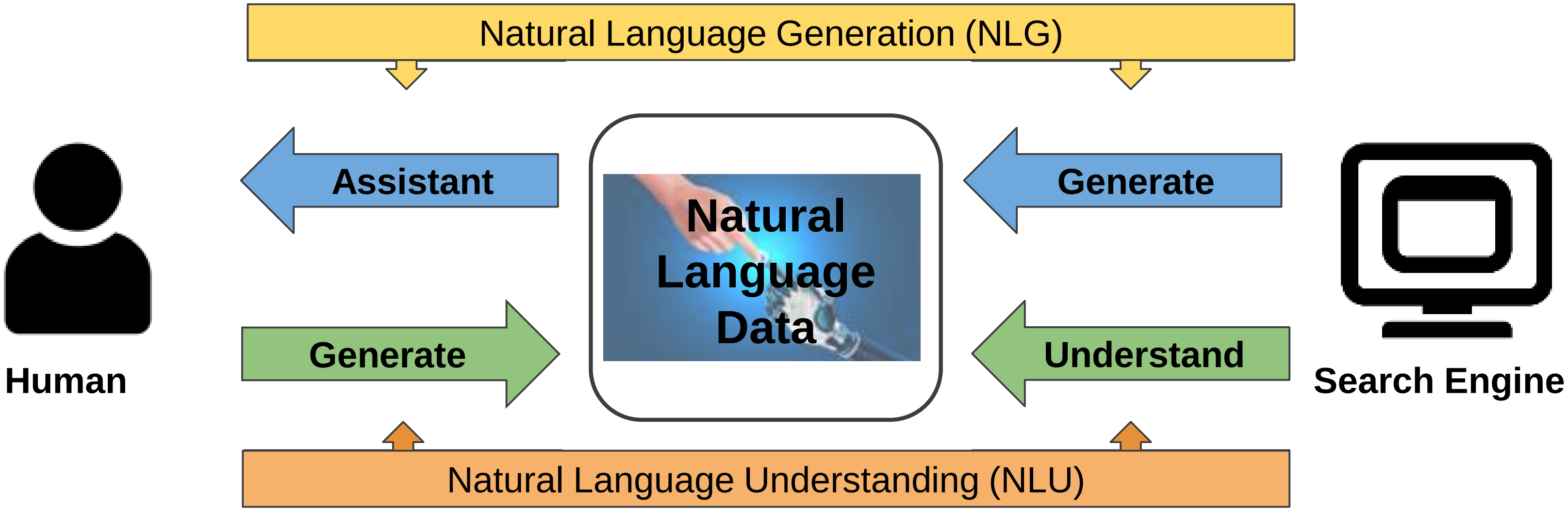
- **Understand Documents**
  - Posts, Reviews, Comments
  - Synonyms, Facets, Semantics, etc.

Best Areas to Stay in Paris Safely

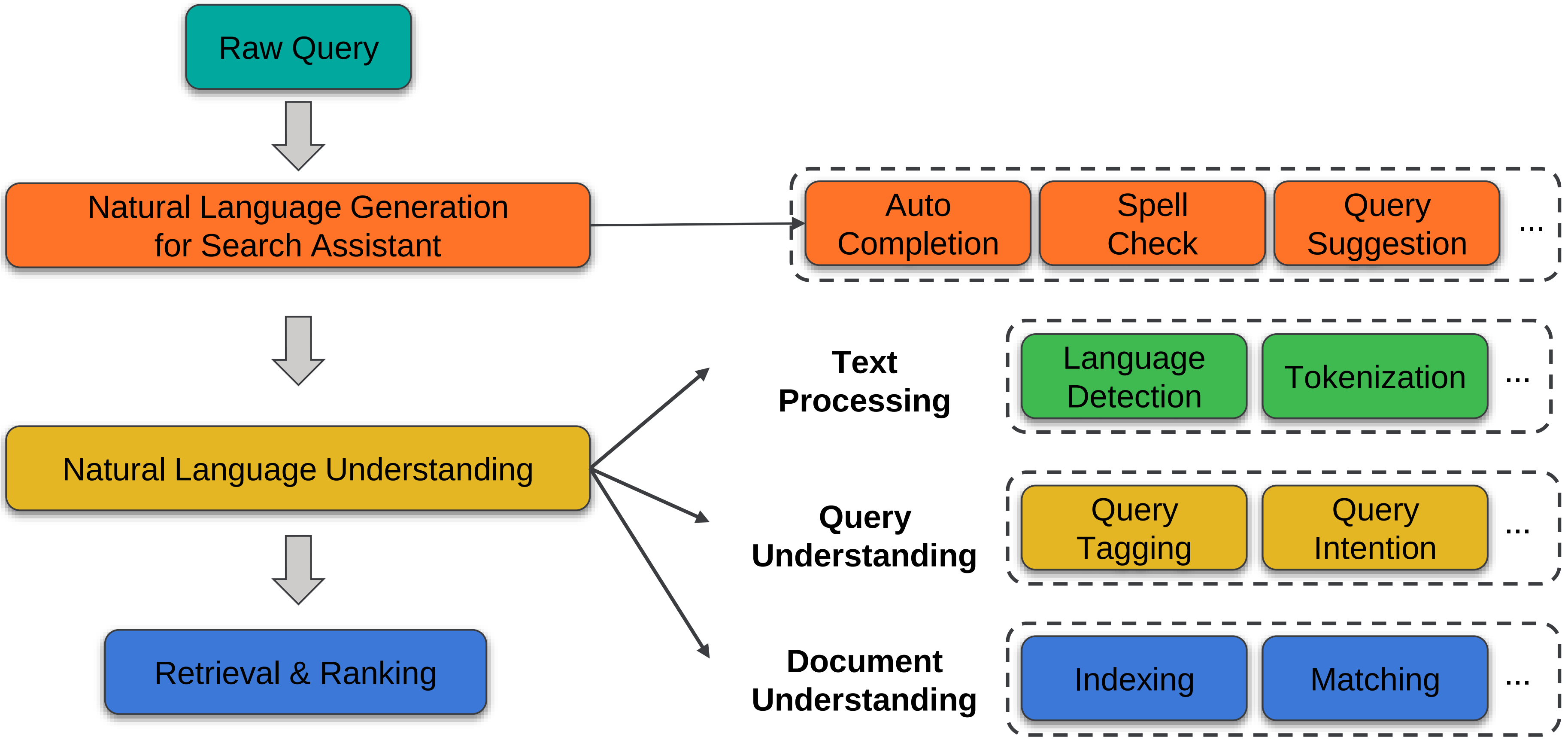
- **Matching**
  - Query & Doc Matching for Retrieval & Ranking

- **Search Assistant**
  - Query Reformulation

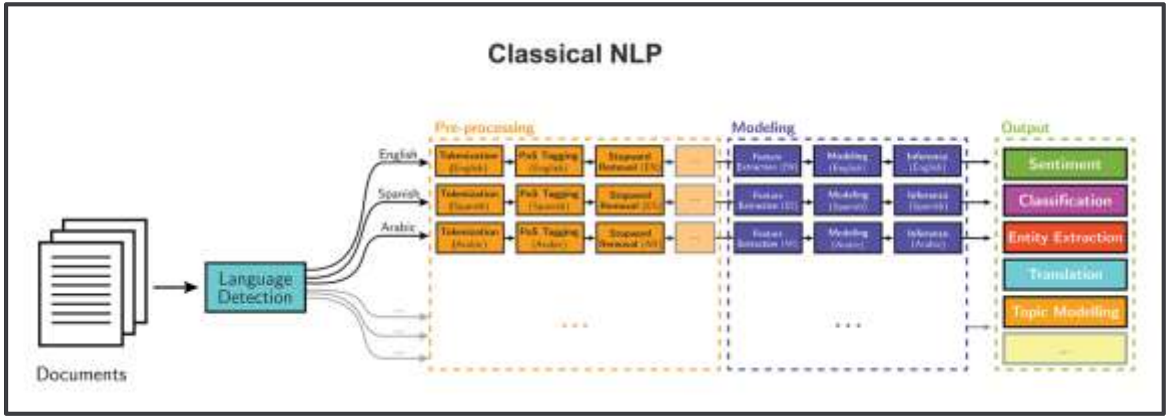
Introduction - NLP in Search Systems



Natural Language Processing in Search Ecosystem



Introduction - Deep Learning for NLP



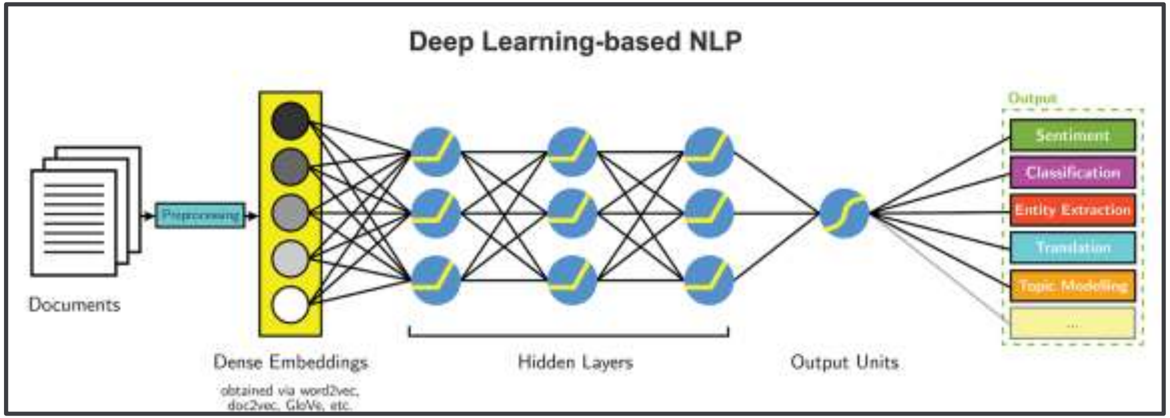


Figure source: Elvis, 2018, https://medium.com/dair-ai/deep-learning-for-nlp-an-overview-of-recent-trends-d0d8f40a776d

Opportunities - Deep Learning for NLP in Search Systems

Why Deep Learning?

- **Deep Semantics** from High Dimension and Sparse Data
  - Synonymous, Disambiguation, etc.

- **Easy Feature Engineering**
  - Hand Crafted Features V.S. Auto Feature Representation

- **Model Flexibility**
  - Model end-to-end process
  - Various NN components to model and cooperate systematically

- **Multi-level Feature Representation**
  - Hierarchical Representations
    character -> token -> word -> phrase -> sentence

[Young et. al. 2018]

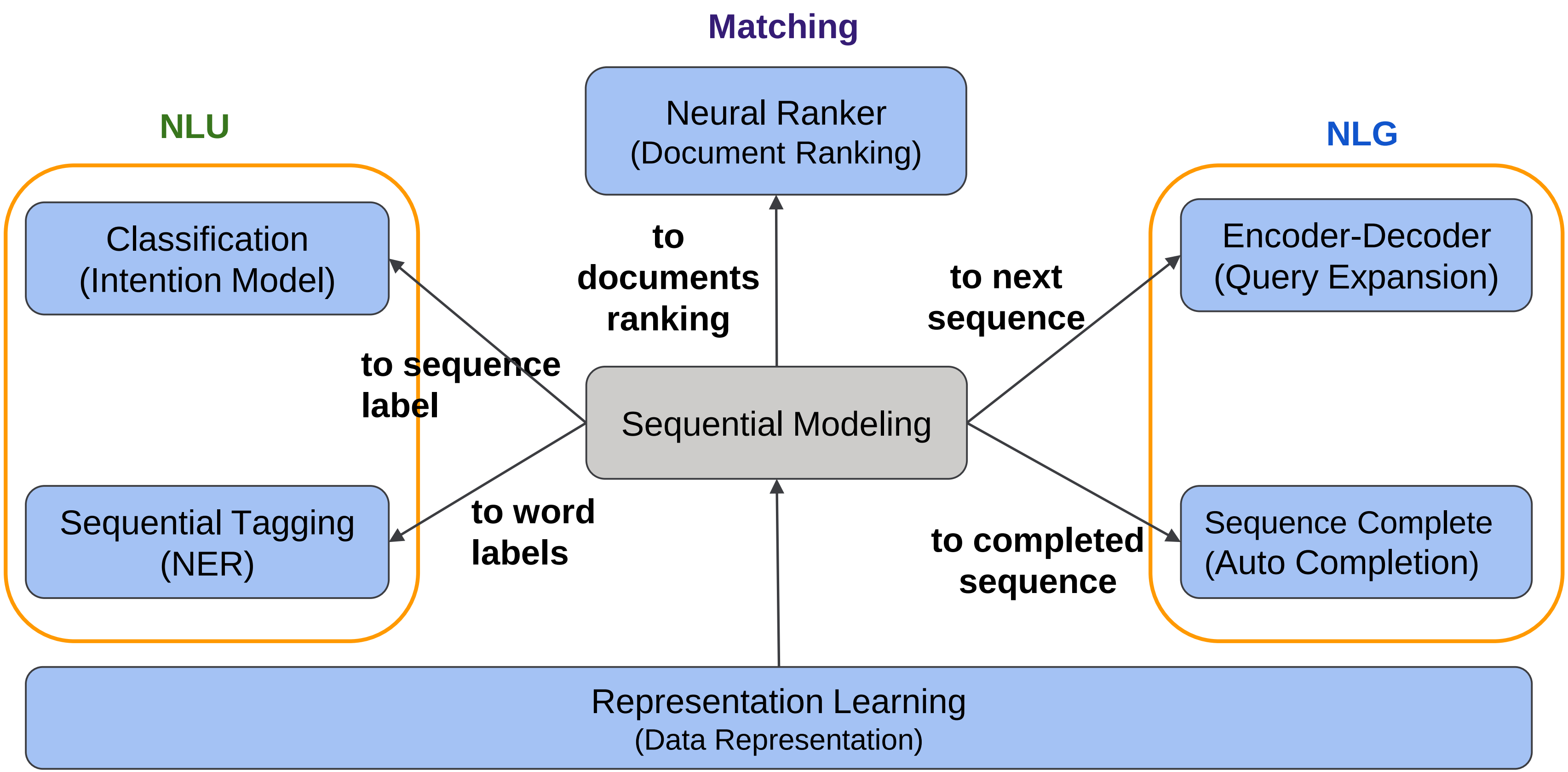
Agenda

1. Introduction
2. **Deep Learning for Natural Language Processing**
3. Deep NLP in Search Systems
4. Real World Examples

Deep Learning for Natural Language Processing

Huiji Gao

Deep Learning for Natural Language Processing

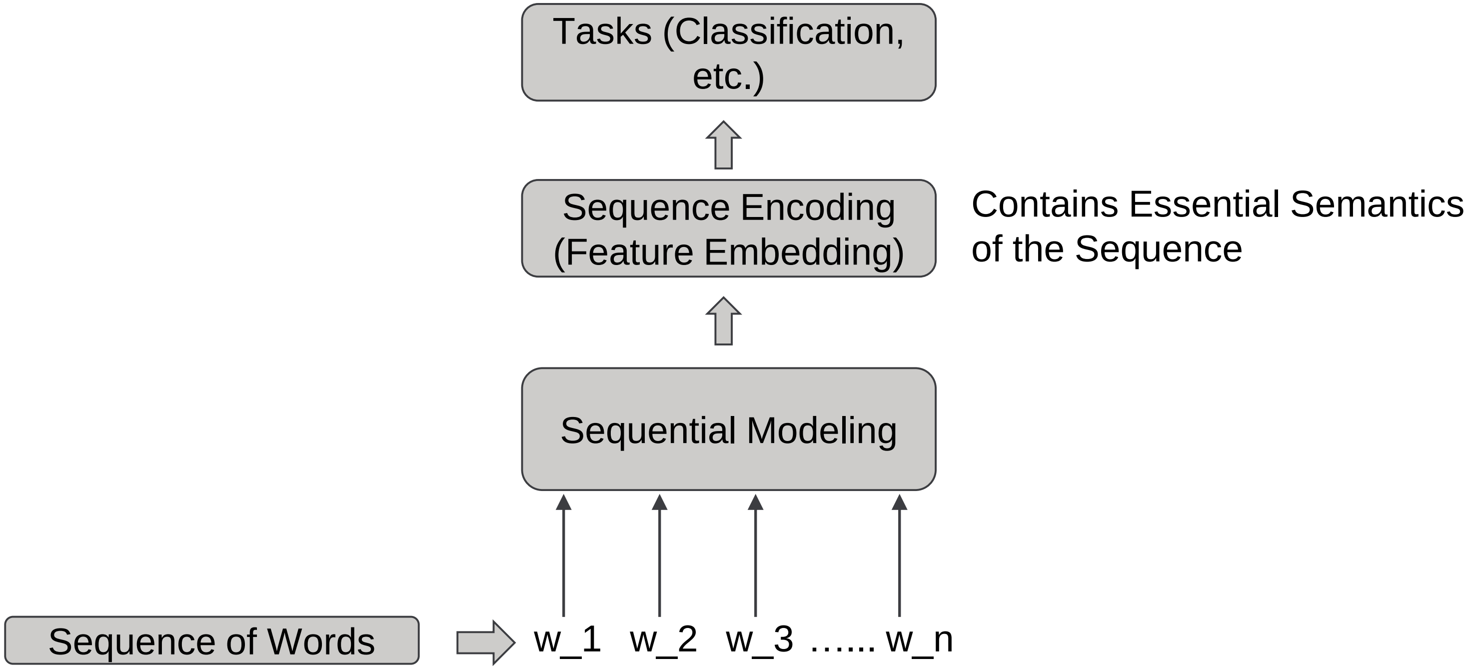


Deep Learning for Natural Language Processing

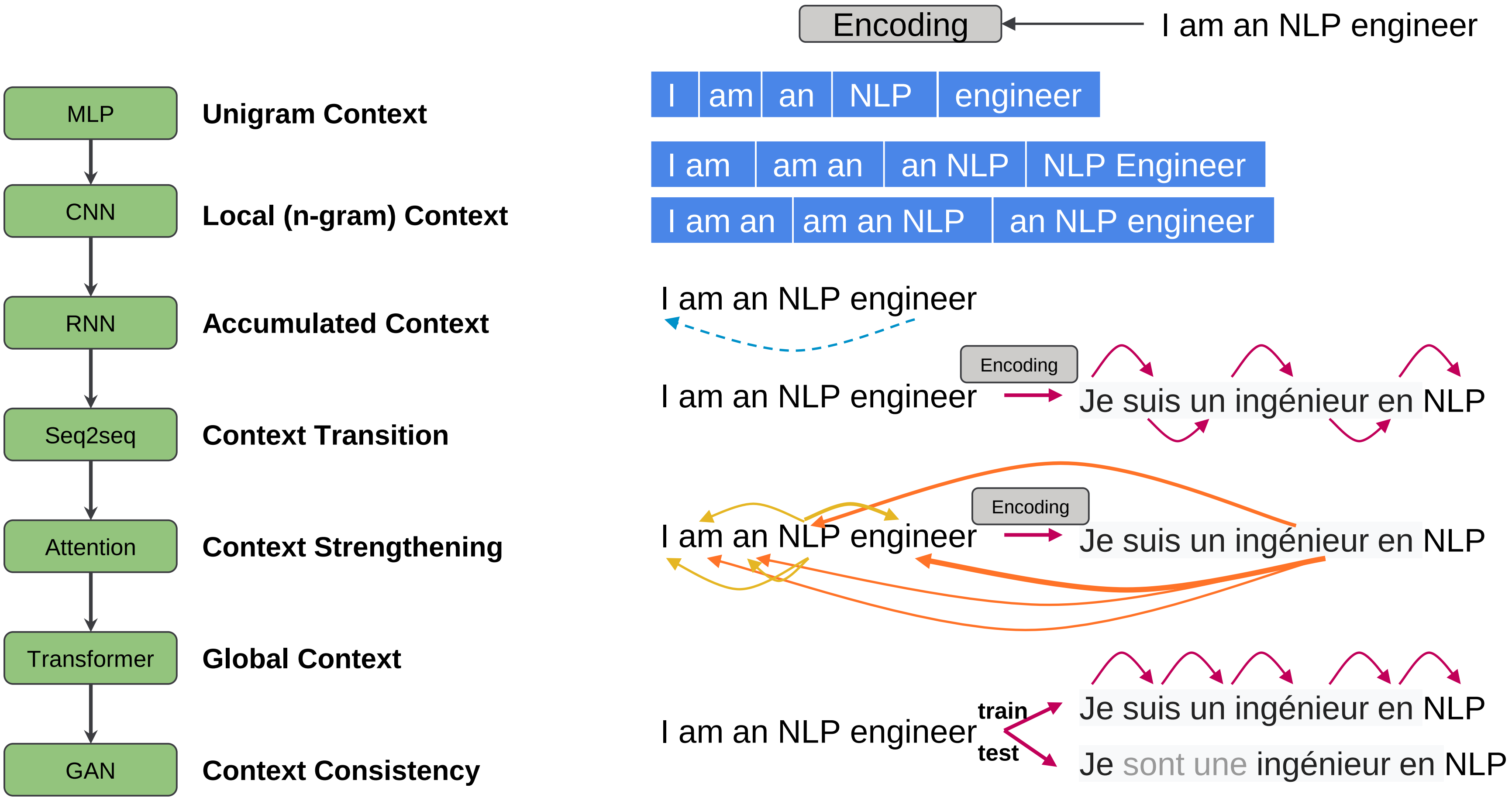
- **Sequential Modeling on Semantic Context**
- Representation Learning for Data Processing

Deep Learning for Natural Language Processing

- **Sequential Modeling on Semantic Context**



Sequential Modeling on Semantic Context



Multilayer Perceptron (MLP)

MLP	**Unigram Context**
CNN	**Local (n-gram) Context**
RNN	**Accumulated Context**
Seq2seq	**Context Transition**
Attention	**Context Strengthening**
Transformer	**Global Context**
GAN	**Context Consistency**

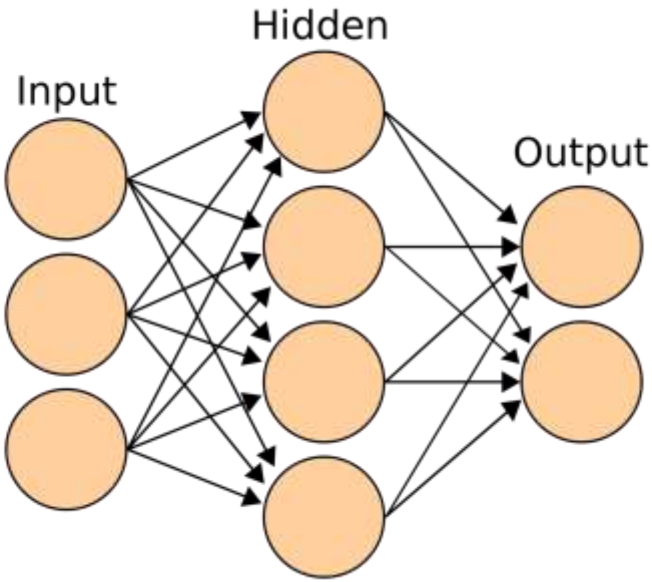
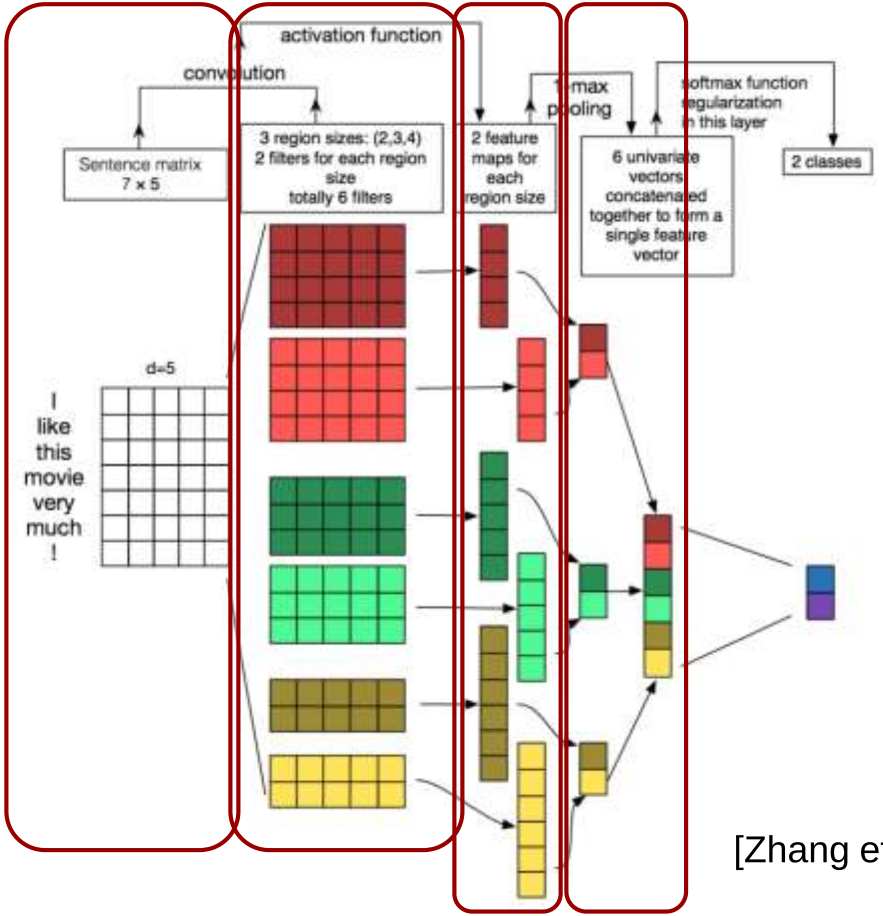


Figure source: https://stackabuse.com/introduction-to-neural-networks-with-scikit-learn/

Convolutional Neural Networks (CNN)

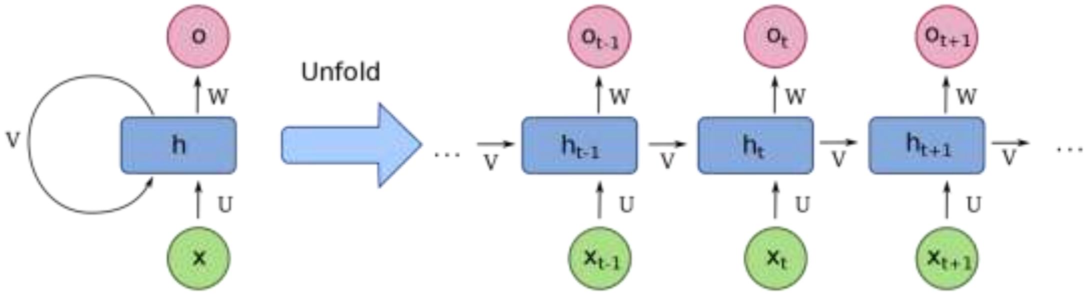
MLP	**Unigram Context**
CNN	**Local (n-gram) Context**
RNN	**Accumulated Context**
Seq2seq	**Context Transition**
Attention	**Context Strengthening**
Transformer	**Global Context**
GAN	**Context Consistency**



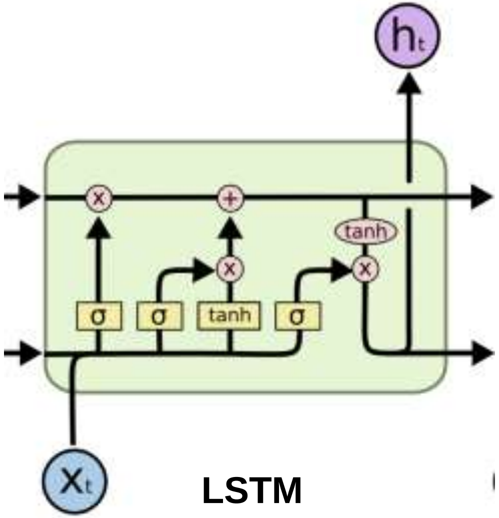
[Zhang et. al. 2015]

Recurrent Neural Networks (RNN)

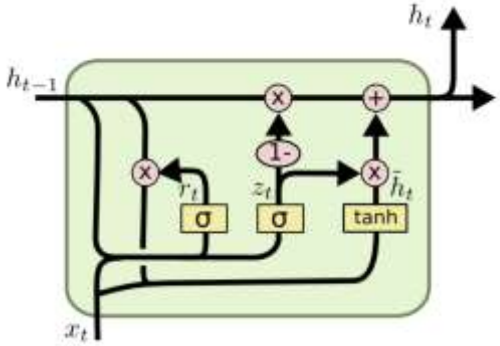
Model	Context
MLP	**Unigram Context**
CNN	**Local (n-gram) Context**
RNN	**Accumulated Context**
Seq2seq	**Context Transition**
Attention	**Context Strengthening**
Transformer	**Global Context**
GAN	**Context Consistency**



[LeCun et. al. 2015]



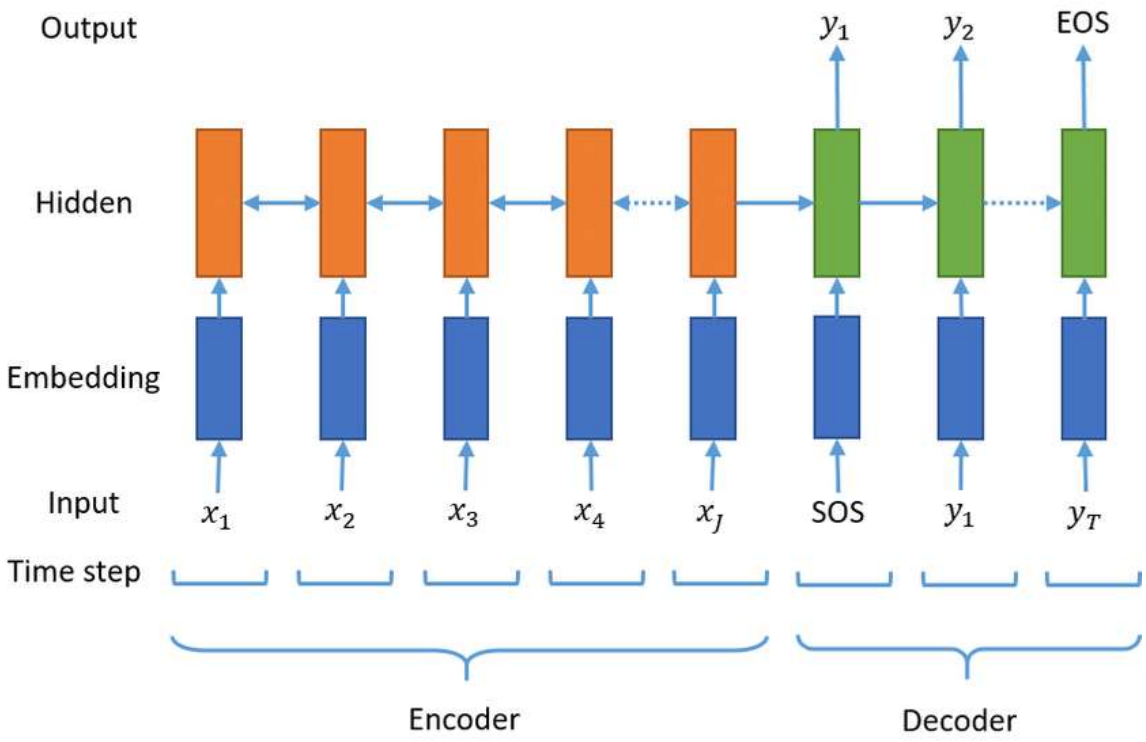
LSTM



GRU

Sequence to Sequence (Encoder - Decoder)

MLP	**Unigram Context**
CNN	**Local (n-gram) Context**
RNN	**Accumulated Context**
Seq2seq	**Context Transition**
Attention	**Context Strengthening**
Transformer	**Global Context**
GAN	**Context Consistency**



[Shi et. al. 2018]

Attention Mechanisms

MLP — **Unigram Context**

CNN — **Local (n-gram) Context**

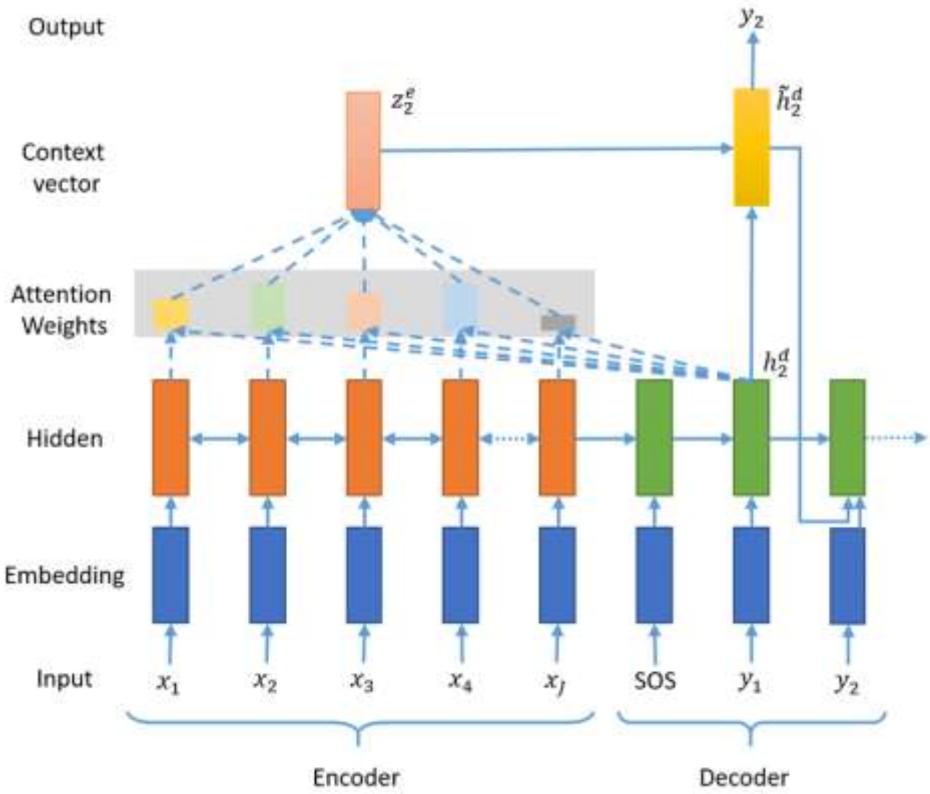
RNN — **Accumulated Context**

Seq2seq — **Context Transition**

Attention — **Context Strengthening**

Transformer — **Global Context**

GAN — **Context Consistency**

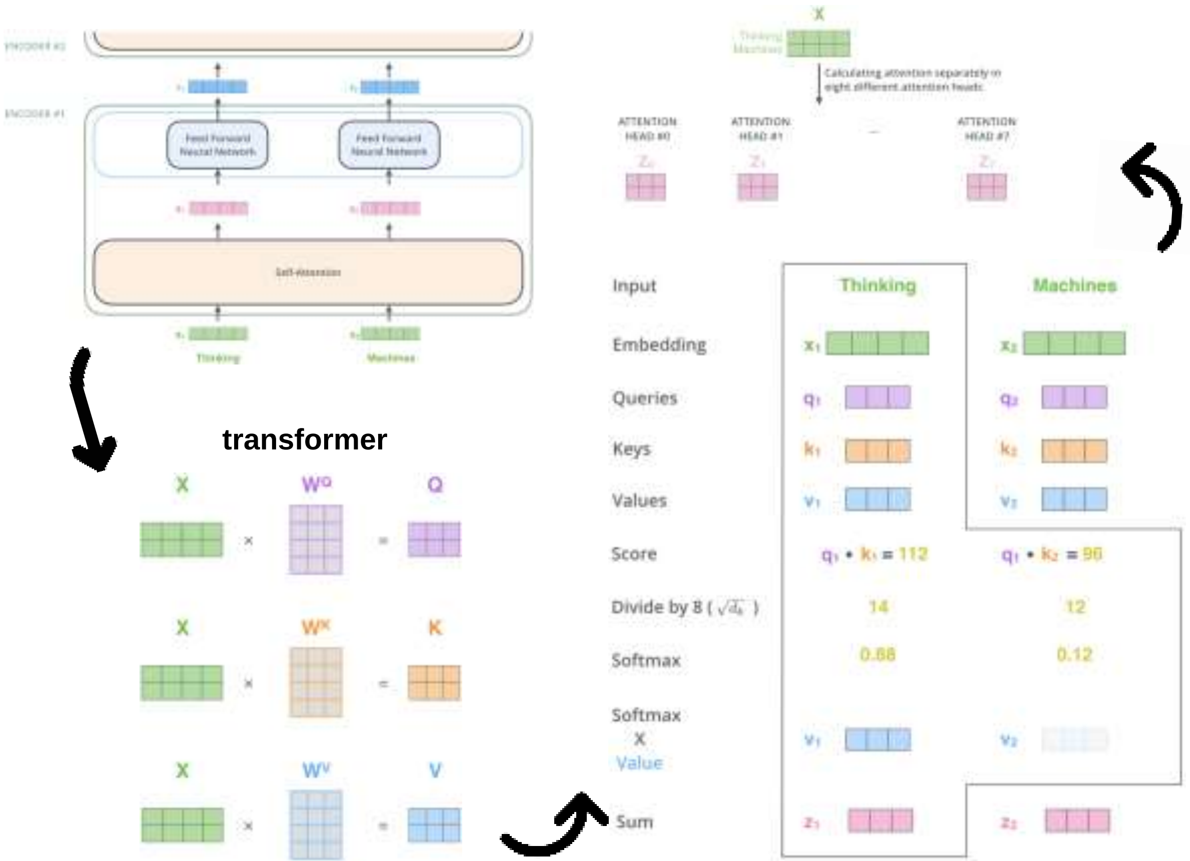


[Shi et. al. 2018]

Transformer

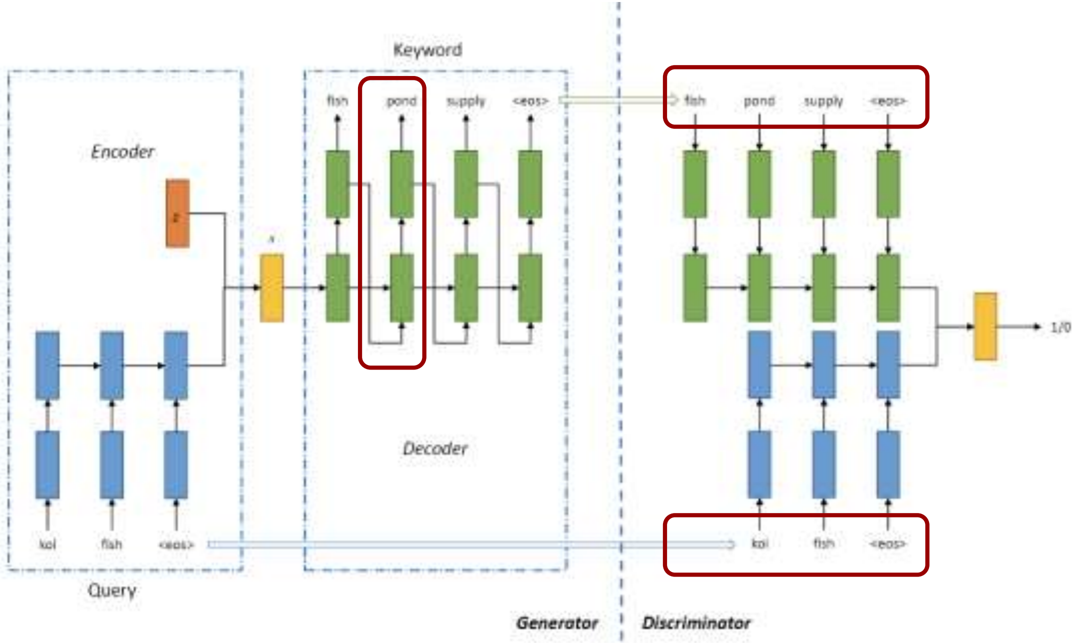
Novelty: Eliminate Recurrence

MLP	**Unigram Context**
CNN	**Local (n-gram) Context**
RNN	**Accumulated Context**
Seq2seq	**Context Transition**
Attention	**Context Strengthening**
Transformer	**Global Context**
GAN	**Context Consistency**



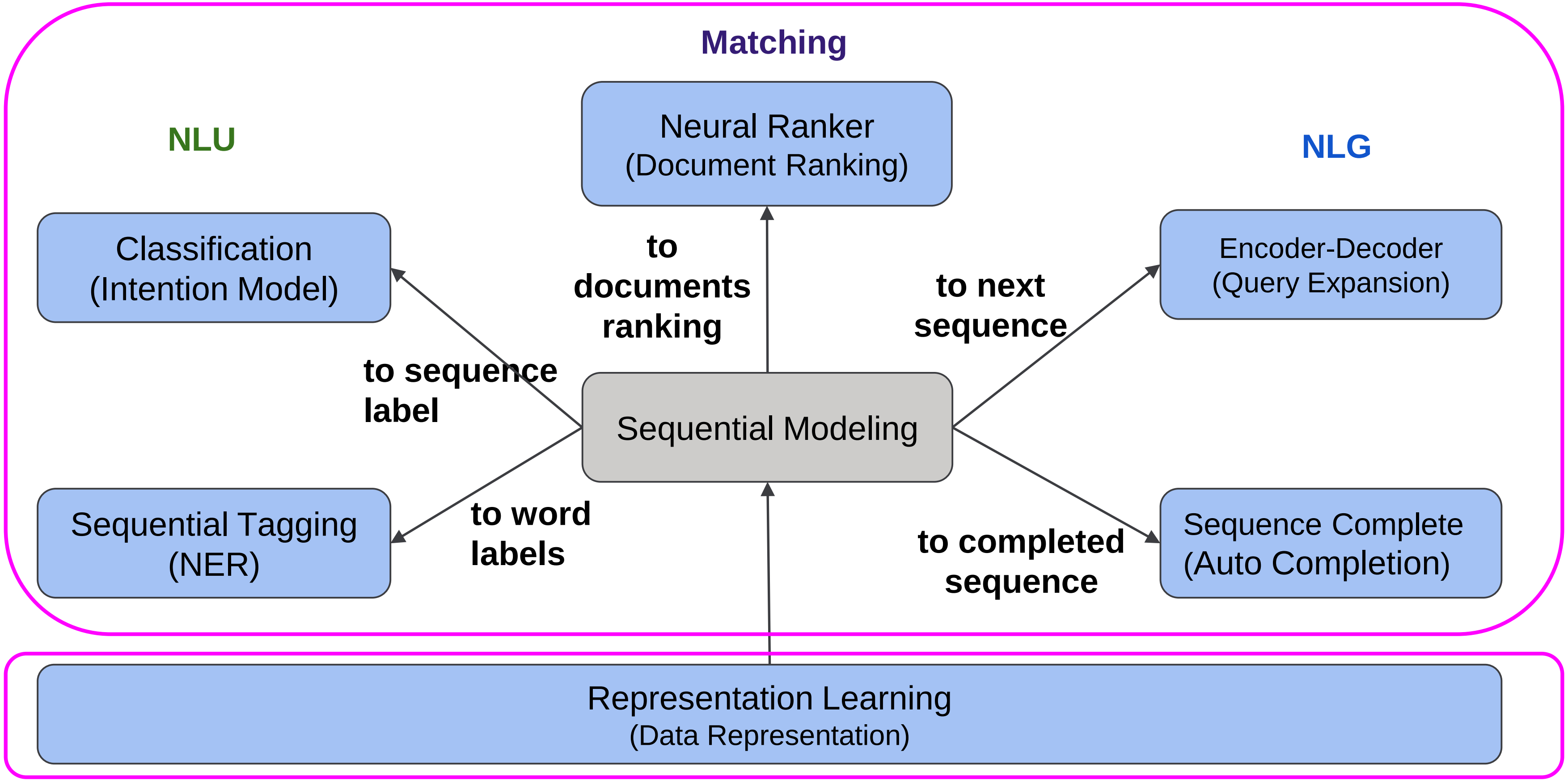
Generative Adversarial Networks (GANs)

MLP	**Unigram Context**
CNN	**Local (n-gram) Context**
RNN	**Accumulated Context**
Seq2seq	**Context Transition**
Attention	**Context Strengthening**
Transformer	**Global Context**
GAN	**Context Consistency**



[Lee, et al., KDD 2018]

Deep Learning for Natural Language Processing

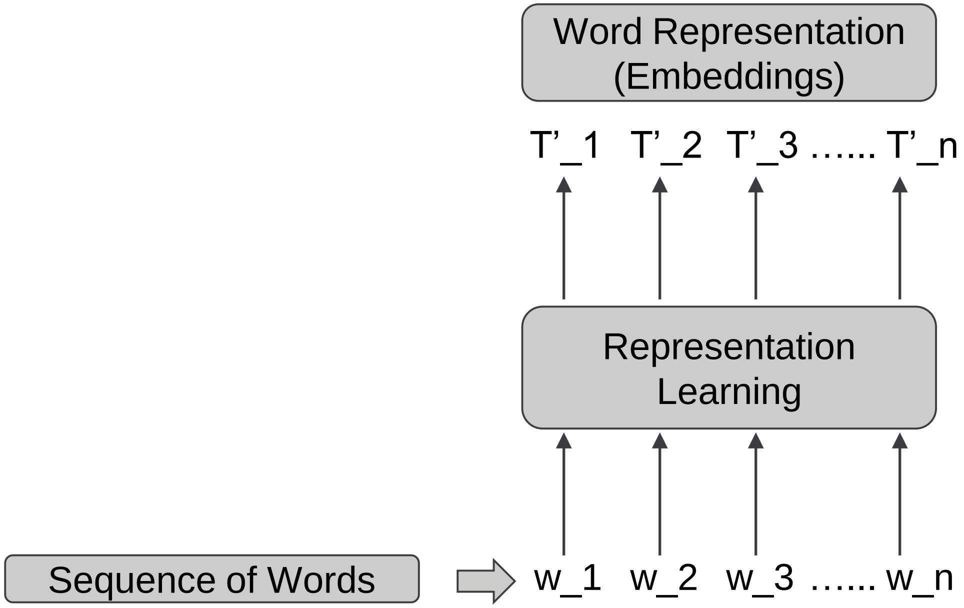


Deep Learning for Natural Language Processing

- Sequential Modeling on Semantic Context
- **Representation Learning for Data Processing**

Deep Learning for Natural Language Processing

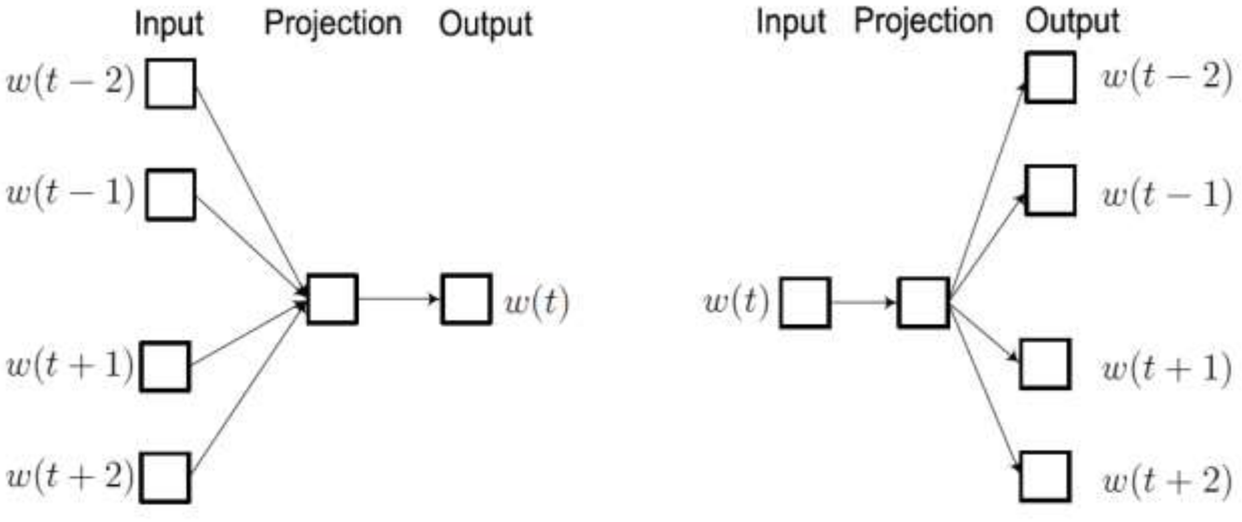
- **Representation Learning**



Representation Learning

Word2Vec

Static Word Embedding



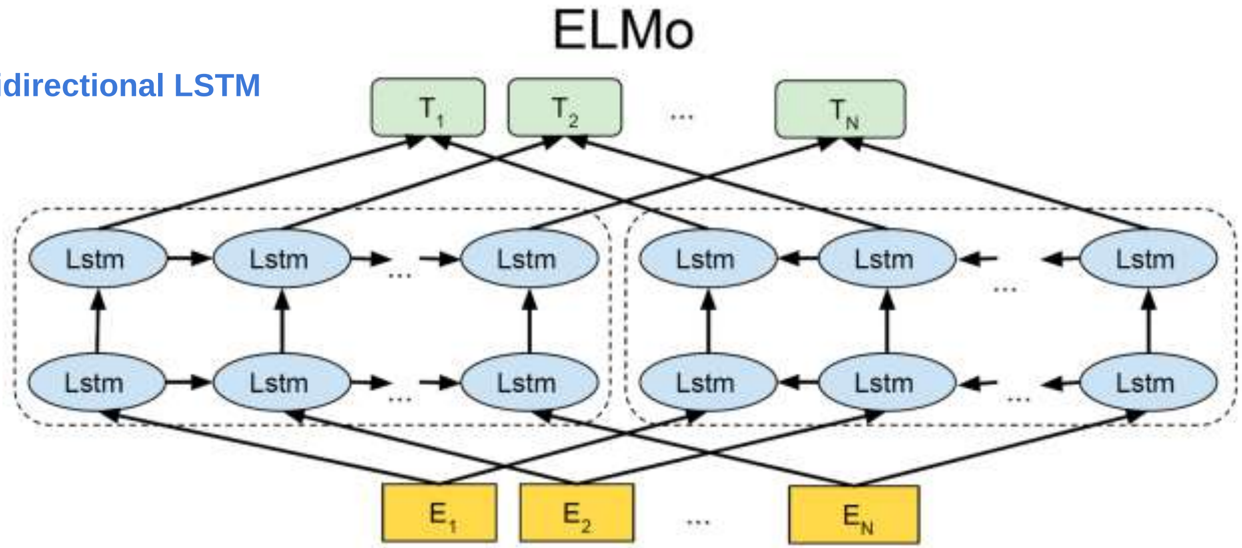
(a) CBOW. (b) Skip-gram.

[Mikolov et. al. 2013]

Representation Learning

Word2Vec — **Static Word Embedding**

ELMO — **Contextual Word Embedding** - Bidirectional LSTM



[Peters et. al. 2018]

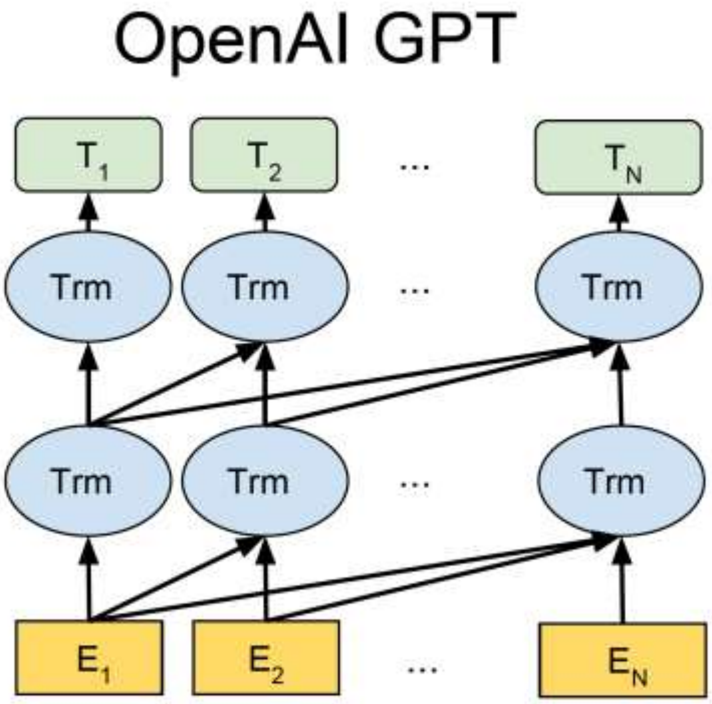
Representation Learning

Word2Vec → ELMO → GPT

Static Word Embedding (Word2Vec)

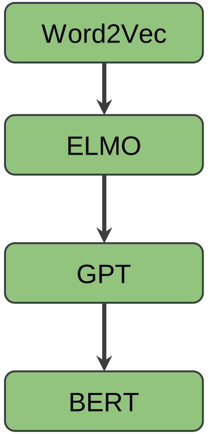
Contextual Word Embedding (ELMO) **- Bidirectional LSTM**

Contextual Word Embedding (GPT) **- One Direction Transformer**



[Radford et. al. 2018]

Representation Learning



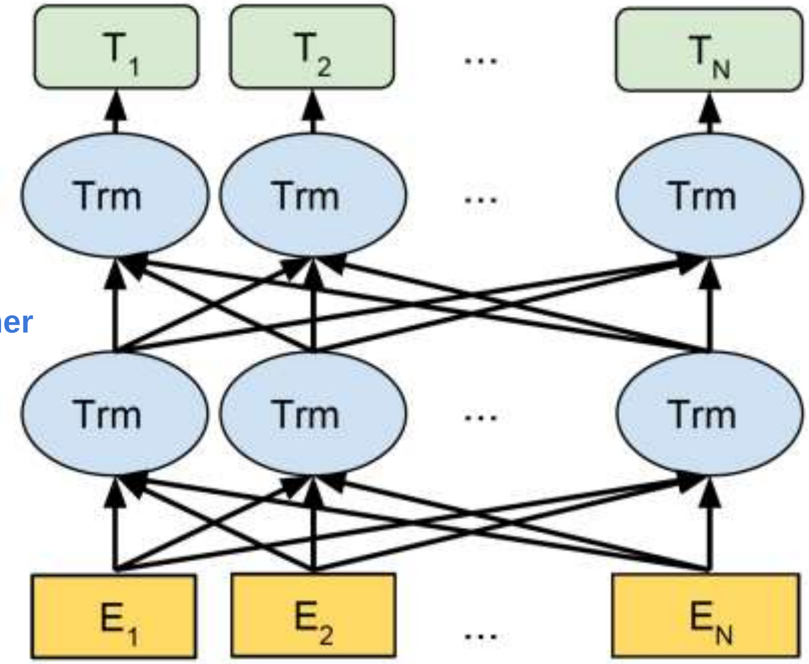
Static Word Embedding

Contextual Word Embedding - Bidirectional LSTM

Contextual Word Embedding - One Direction Transformer

Contextual Word Embedding - Bidirectional Transformer with Multitask Learning

BERT



[Devlin et. al. 2018]

Representation Learning

Word2Vec → ELMO → GPT → BERT → XLNET

Word2Vec — **Static Word Embedding**

ELMO — **Contextual Word Embedding** - **Bidirectional LSTM**

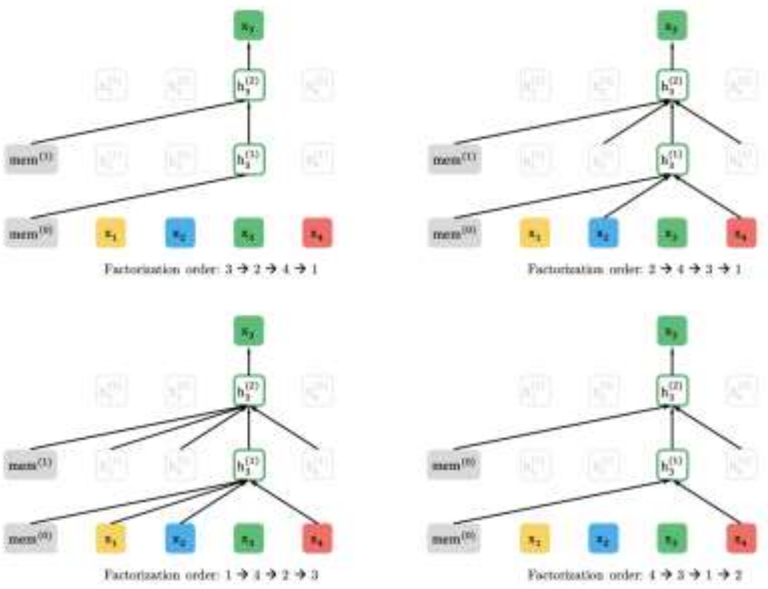
GPT — **Contextual Word Embedding** - **One Direction Transformer**

BERT — **Contextual Word Embedding** - **Bidirectional Transformer with Multitask Learning**

XLNET — **Contextual Word Embedding** - **Bidirectional, Learn correlations among masked tokens**

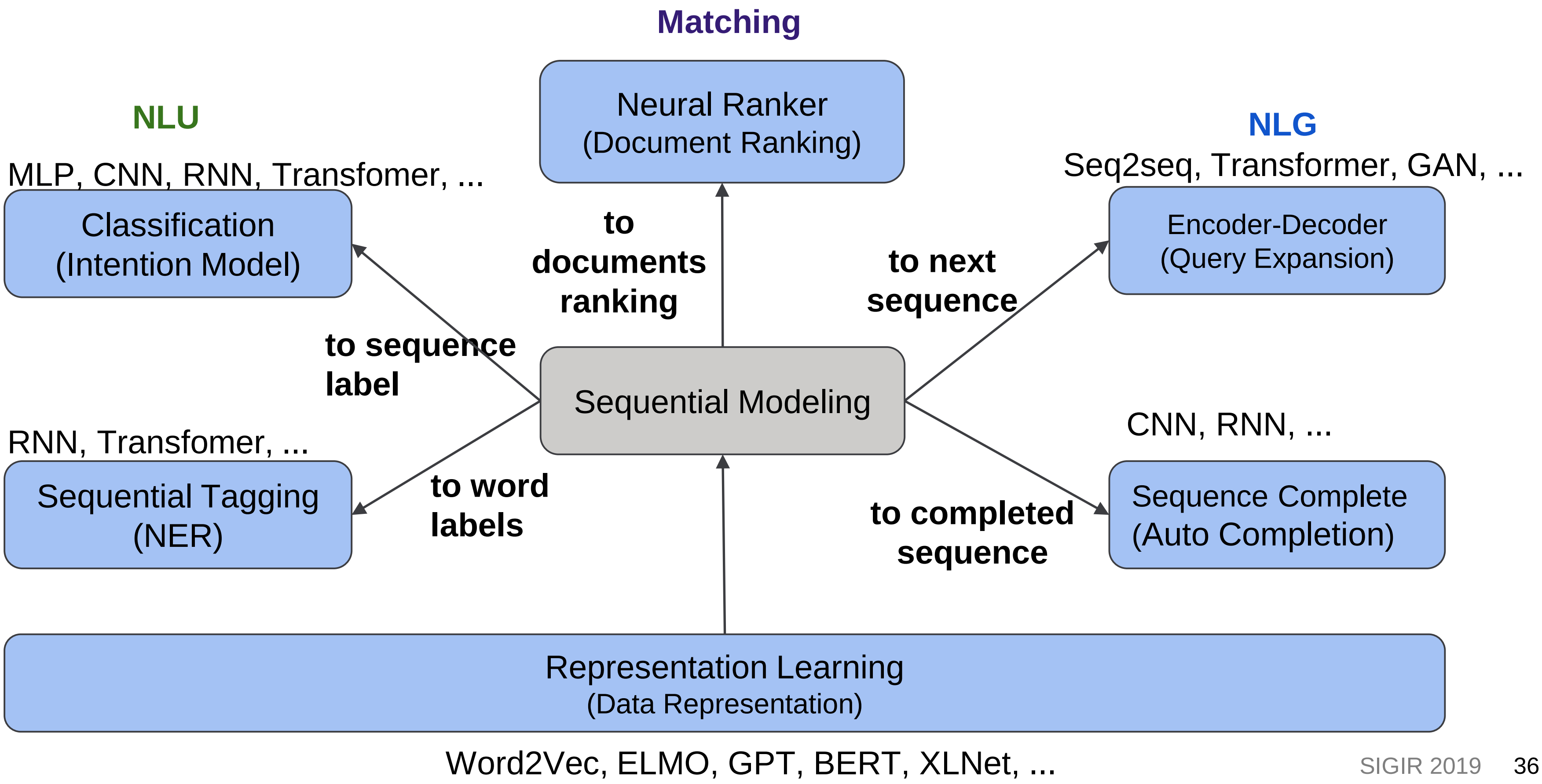
Pre-trained NLP Model

XLNet



[Yang et. al. 2018]

Deep Learning for Natural Language Processing



References - Deep Learning for Natural Language Processing

- [Devlin et. al. 2018] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805, 2018
- [LeCun et. al, 2015] Deep learning, Nature, vol. 521, no. 7553, pp. 436–444, 2015.
- [Lee et. al, 2018] Rare query expansion through generative adversarial networks in search advertising, In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 500-508. ACM, 2018.
- [Mikolov et. al. 2013] Distributed representations of words and phrases and their compositionality, in Advances in neural information processing systems, 2013
- [Mikolov et. al. 2013] Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781, 2013.
- [Peters et. al. 2018] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In NAACL.
- [Radford et.al. 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.
- [Robinson, 2018] https://stackabuse.com/introduction-to-neural-networks-with-scikit-learn, 2018.
- [Shi et. al, 2018] Neural Abstractive Text Summarization with Sequence-to-Sequence Models, arXiv preprint arXiv:1812.02303v2, 2018.
- [Yang et. al. 2019] XLNet: Generalized Autoregressive Pretraining for Language Understanding, arXiv preprint arXiv:1906.08237v1, 2019
- [Young et. al. 2018] Recent Trends in Deep Learning Based Natural Language Processing, arXiv preprint arXiv:1708.02709v8, 2018
- [Zhang et. al. 2015] A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification, arXiv preprint arXiv:1510.03820, 2015.

Agenda

1. Introduction
2. Deep Learning for Natural Language Processing
3. **Deep NLP in Search Systems**
4. Real World Examples

Deep NLP in Search Systems

- Language Understanding

Jun Shi

Deep NLP in Search Systems

- **Language Understanding**
  - **Entity Tagging: word level prediction**
  - **Entity Disambiguation: knowledge base entity prediction**
  - **Intent Classification: sentence level prediction**
- Document Retrieval and Ranking
  - Efficient Candidate Retrieval
  - Deep Ranking Models
- Language Generation for Search Assistance
  - Query Suggestion: word-level sequence to sequence
  - Spell Correction: character-level sequence to sequence
  - Auto Complete: partial sequence to sequence

Entity Tagging

- Problem statement
- Traditional statistical models
  - Hidden Markov Model
  - Maximum Entropy Markov Model
  - (Semi-Markov) Conditional Random Field
- Deep learning models
  - Input layer
  - Context layer
  - Decoder layer
- Training dataset
- Evaluation

Entity Tagging - Problem Statement

- A named entity, a word or a phrase that clearly identifies one item from a set of other items that have similar attributes. [Li et. al. 2018]
- Entity tagging (Named Entity Recognition, NER), the process of locating and classifying named entities in text into predefined entity categories.

Washington D.C. was named after George Washington.
LOCATION PERSON

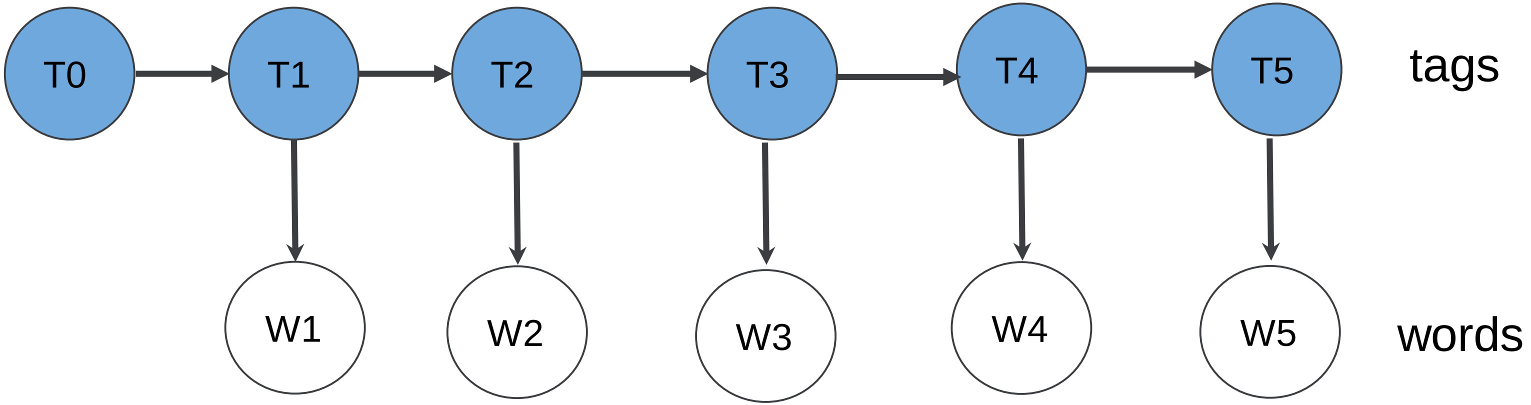
Motivation

- Efficiency
  - Looking only for named entities can speed up search.
- Precision
  - Matching both named entity and tags can increase search precision.
- Quality
  - Ranking retrieved results by considering tags improves search quality.

Traditional Statistical Models

- Generative model
  - Hidden Markov Model [Baum, et. al. 1966]
- Discriminative model
  - Maximum Entropy Markov Model [McCallum, et. al. 2000]
  - (Semi-Markov) Conditional Random Field [Lafferty, et. al. 2001]

Hidden Markov Model

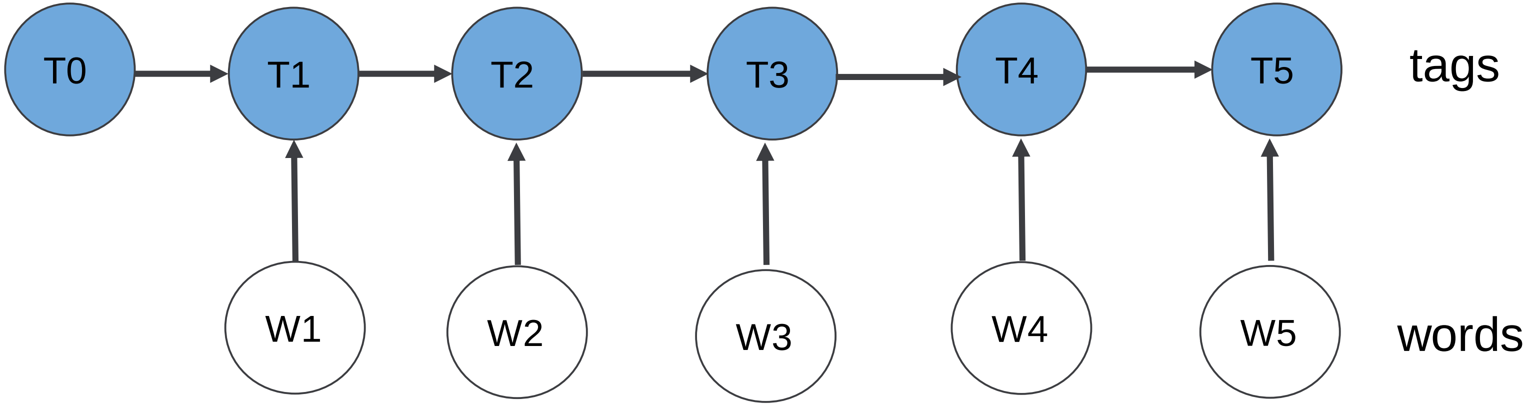


Generative model, model joint probability Pr(T, W) instead of conditional probability Pr(T | W).

$$\Pr(\boldsymbol{T}, \boldsymbol{W}) = \prod_{i=1}^{L} (\Pr(w_i | t_i) \Pr(t_i | t_{i-1}))$$

t_0 is a dummy start state.

Maximum Entropy Markov Model

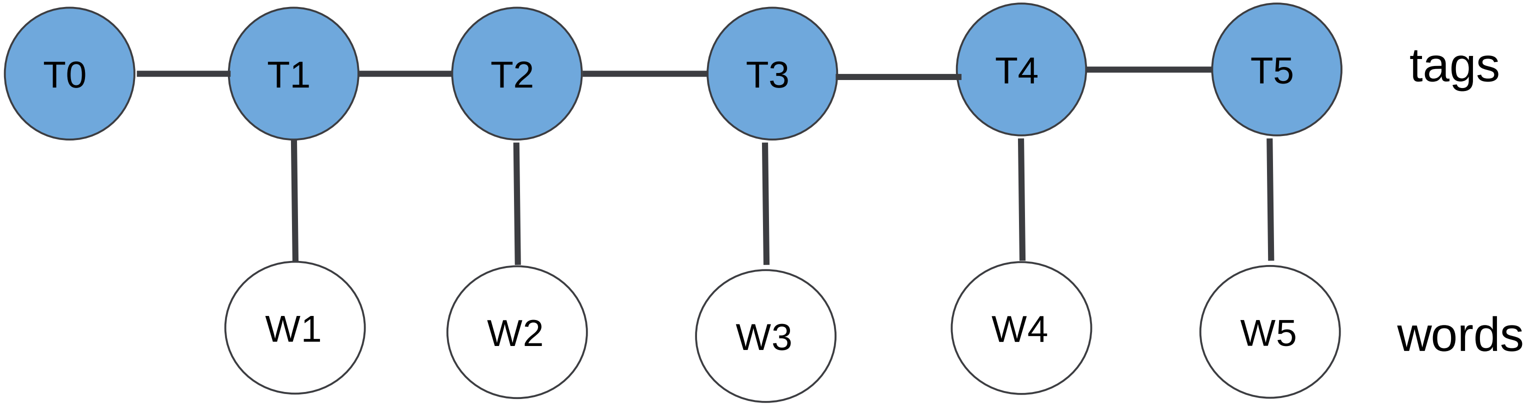


Discriminative model, model conditional probability Pr(T | W) directly.

$$\Pr(\boldsymbol{T}|\boldsymbol{W}) = \prod_{i=1}^{L} \Pr(t_i|t_{i-1}, w_i) = \prod_{i=1}^{L} \frac{\exp(\sum_j \beta_j f_j(t_{i-1}, w_i))}{Z(t_{i-1}, w_i)}$$

t_0 is a dummy start state.

Conditional Random Field

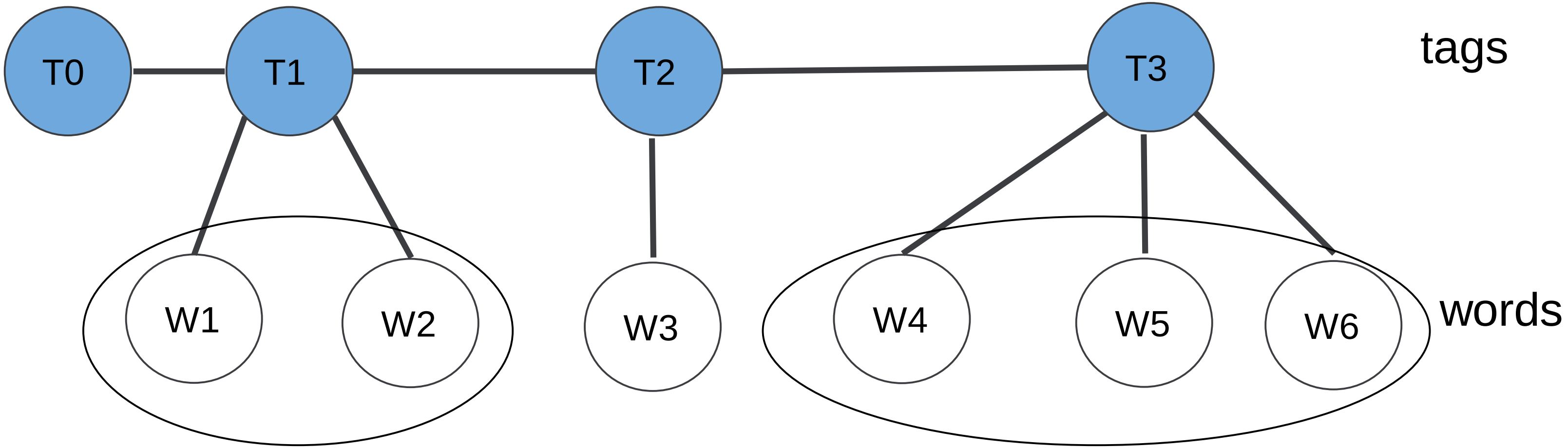


Discriminative model, model conditional probability Pr(T | W) directly.

$$\Pr(\mathbf{T}|\mathbf{W}) = \frac{\prod_{i=1}^{L} \exp(\Sigma_j \beta_j f_j(t_{i-1}, \mathbf{W}))}{Z(\mathbf{T}, \mathbf{W})}$$

t_0 is a dummy start state.

Semi-Markov Conditional Random Field

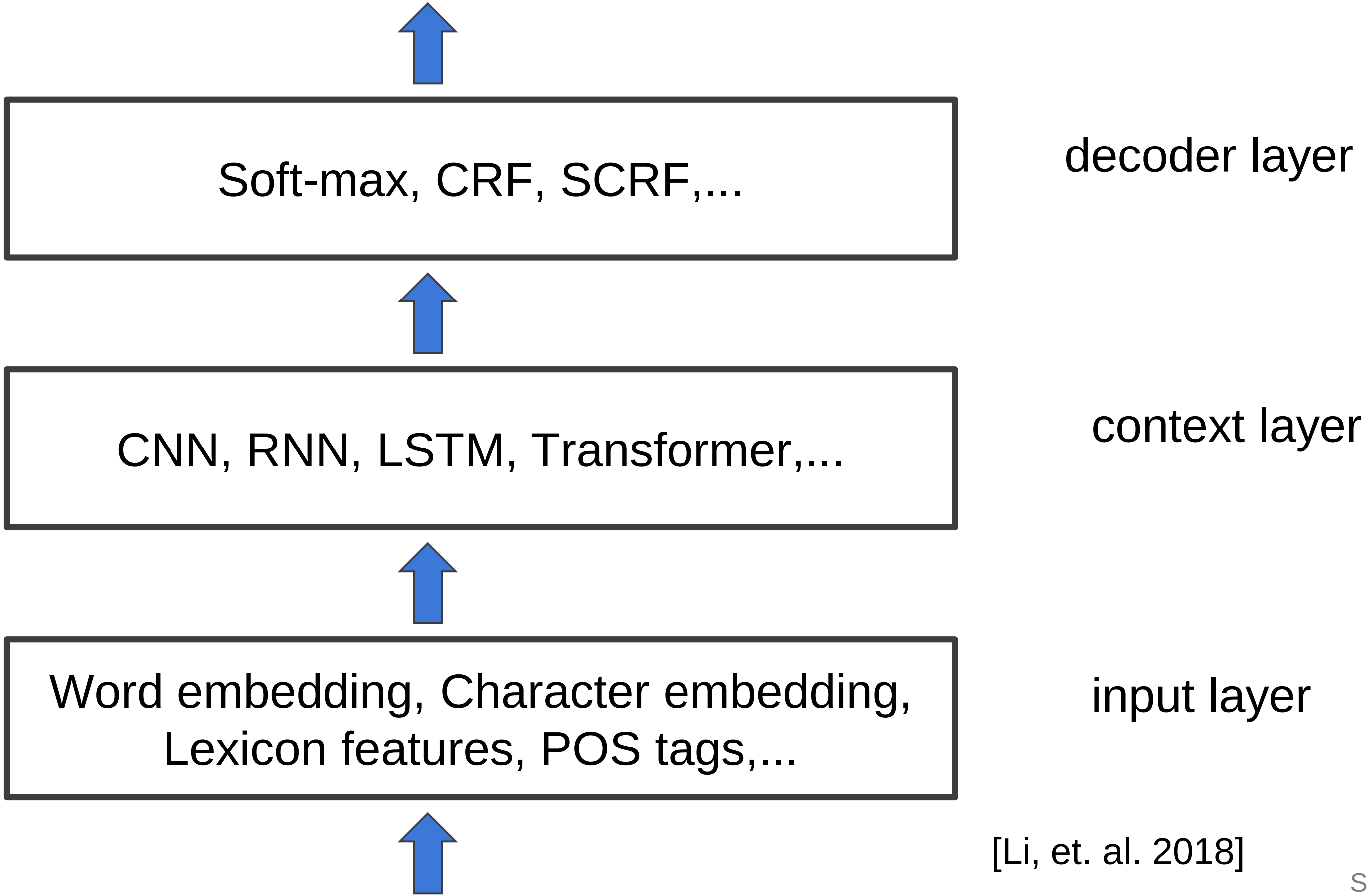


Each tag can correspond to a variable-length phrase.

Summary - Traditional Statistical Models

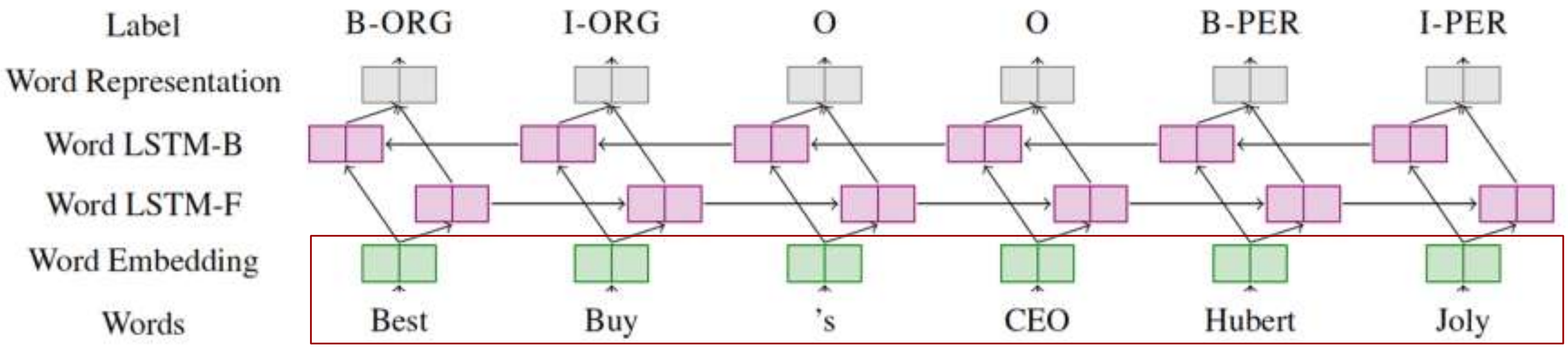
Model	Pros	Cons
Hidden Markov Model	training is simple when states are observed	difficult to include features
Maximum Entropy Markov Model	easy to include features	suffer from “label bias problem” (prefer states with lower number of transitions)
Conditional Random Field	easy to include features, do not suffer from “label bias problem”	training is relatively complex

Deep Learning Tagger Architecture



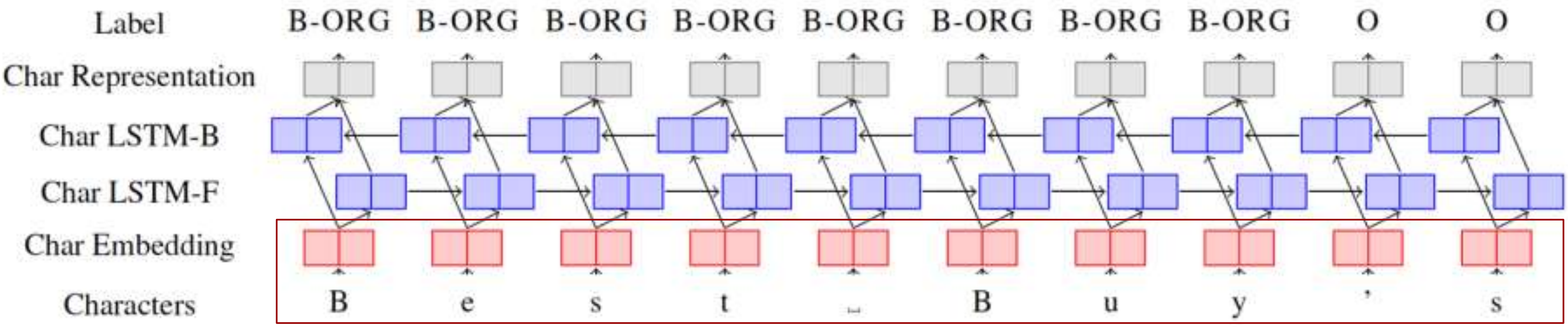
[Li, et. al. 2018]

Input Layer - Word Embedding



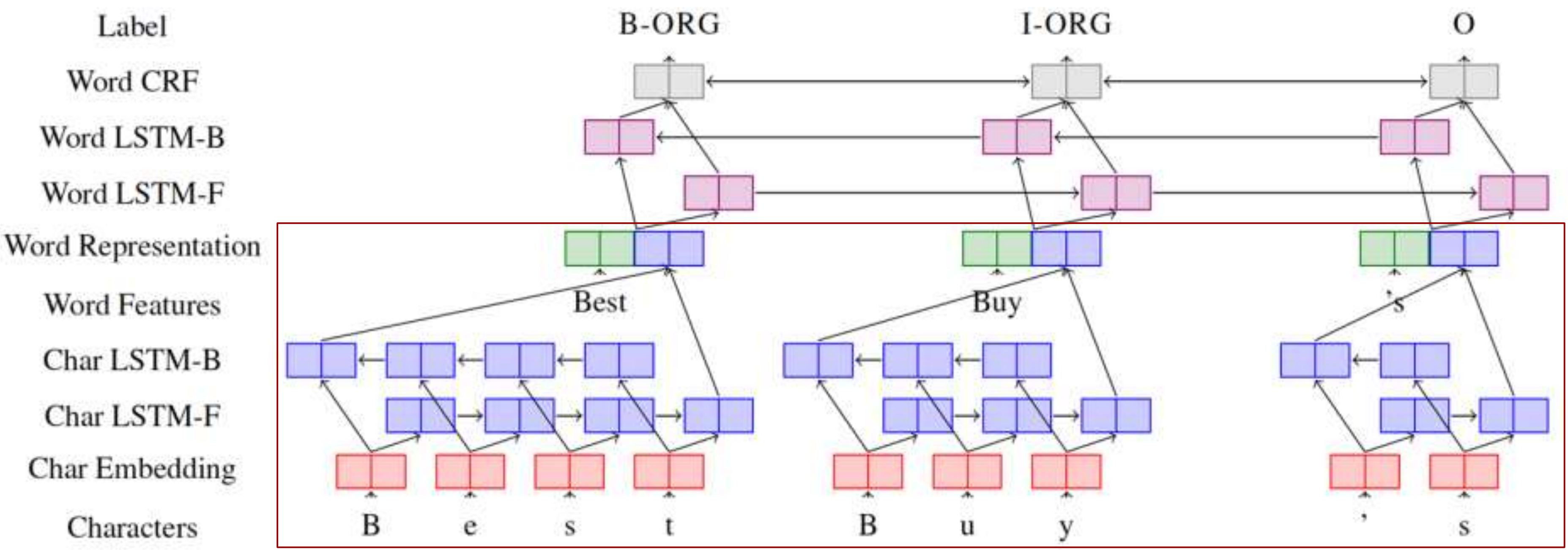
[Yadav, et. al. 2018]

Input Layer - Char Embedding



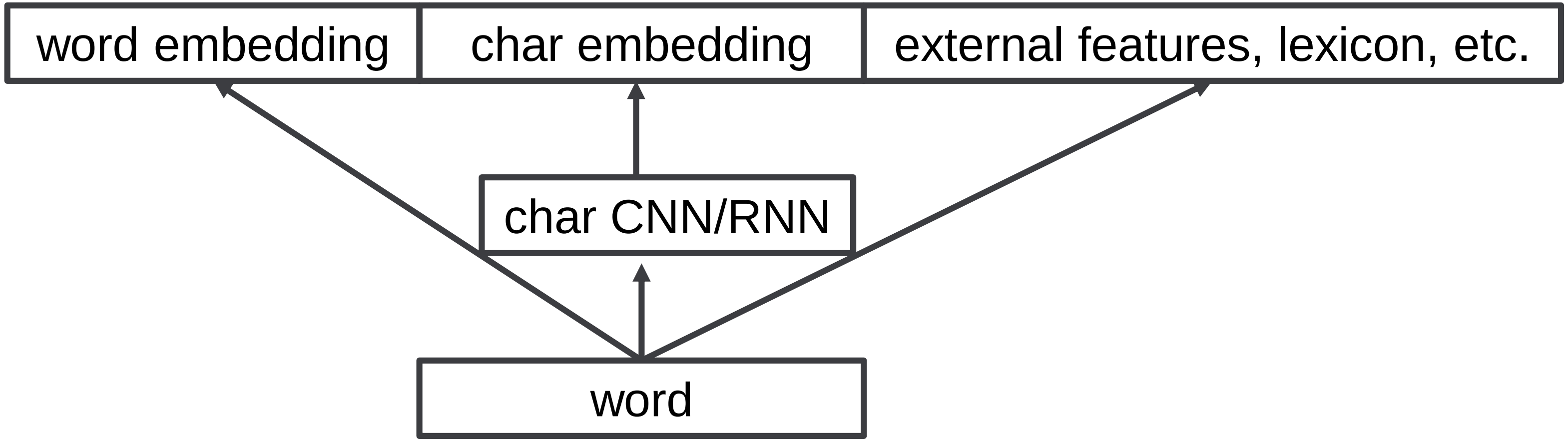
[Yadav, et. al. 2018]

Input Layer - Word and Char Embedding

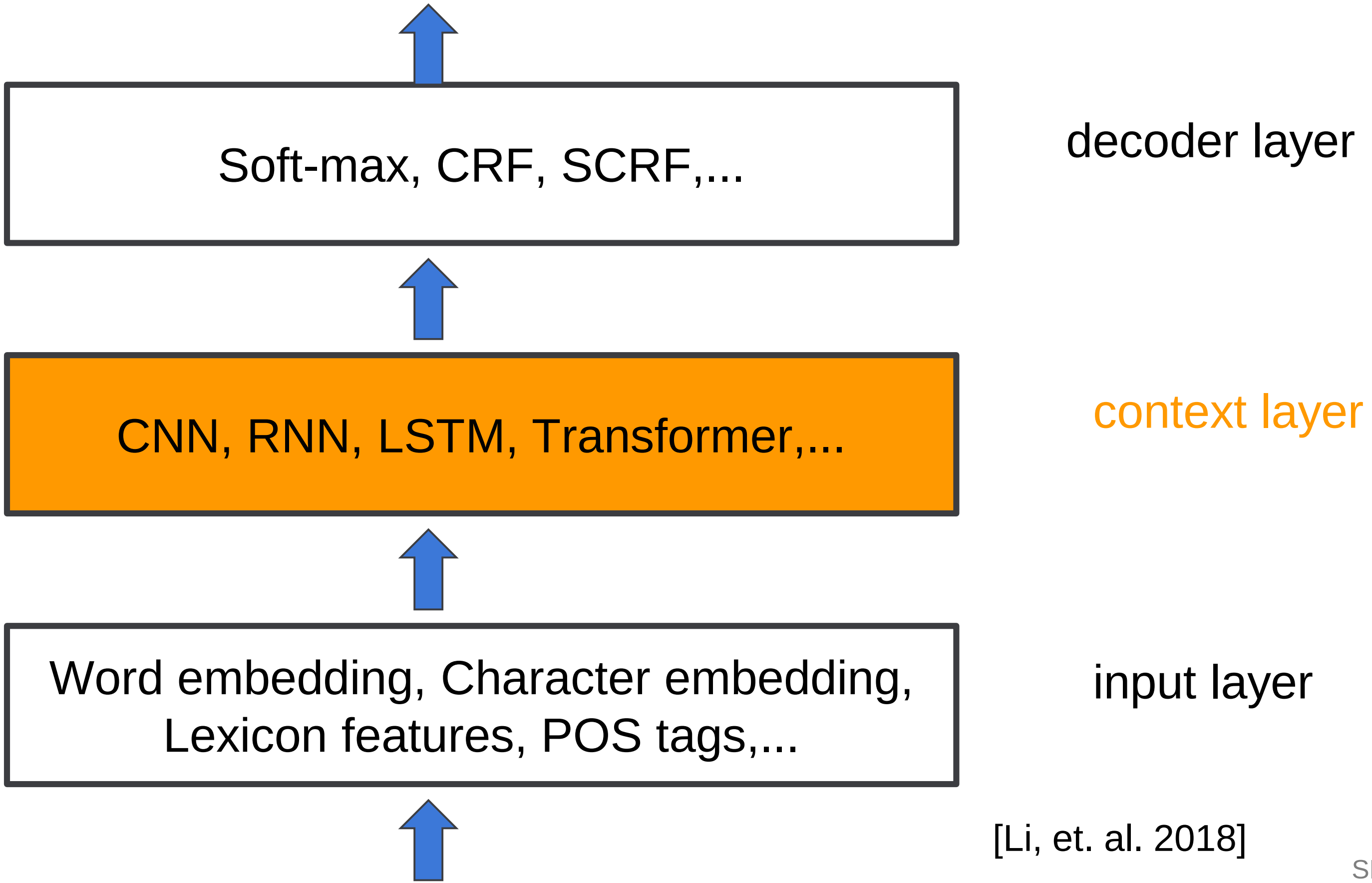


[Yadav, et. al. 2018]

Summary - Input Layer

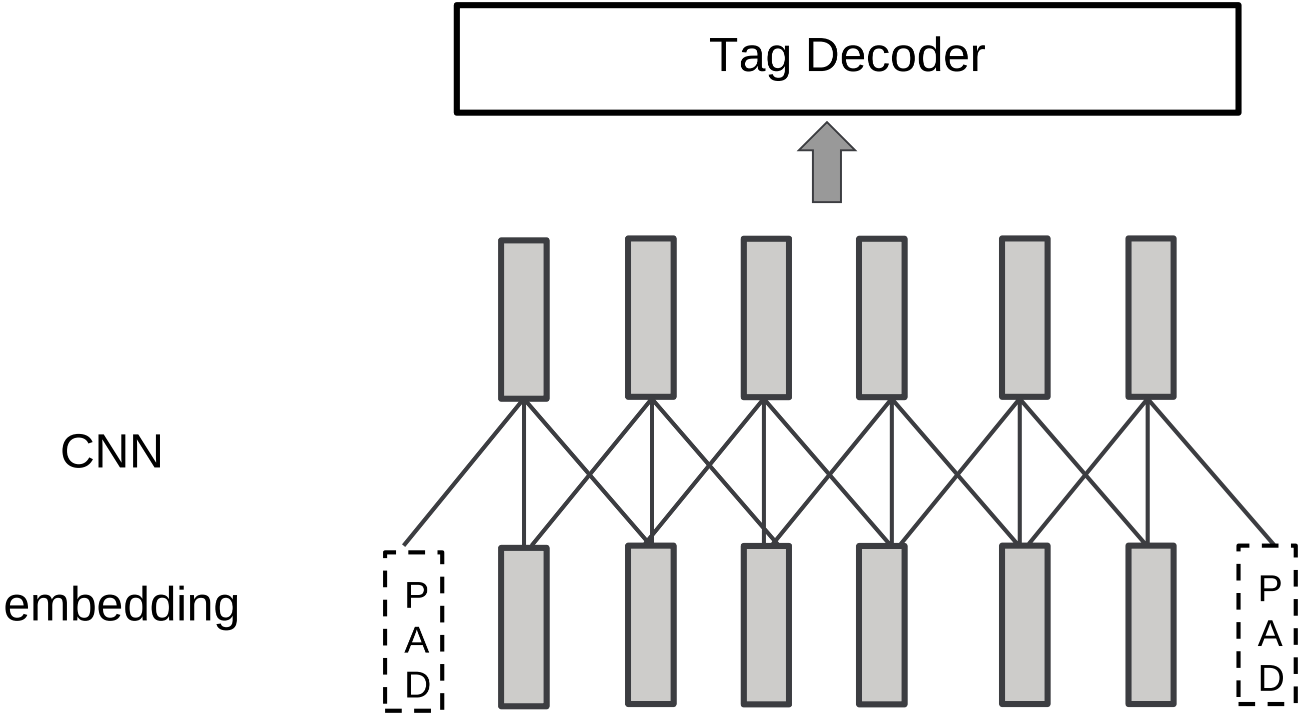


Deep Learning Tagger Architecture



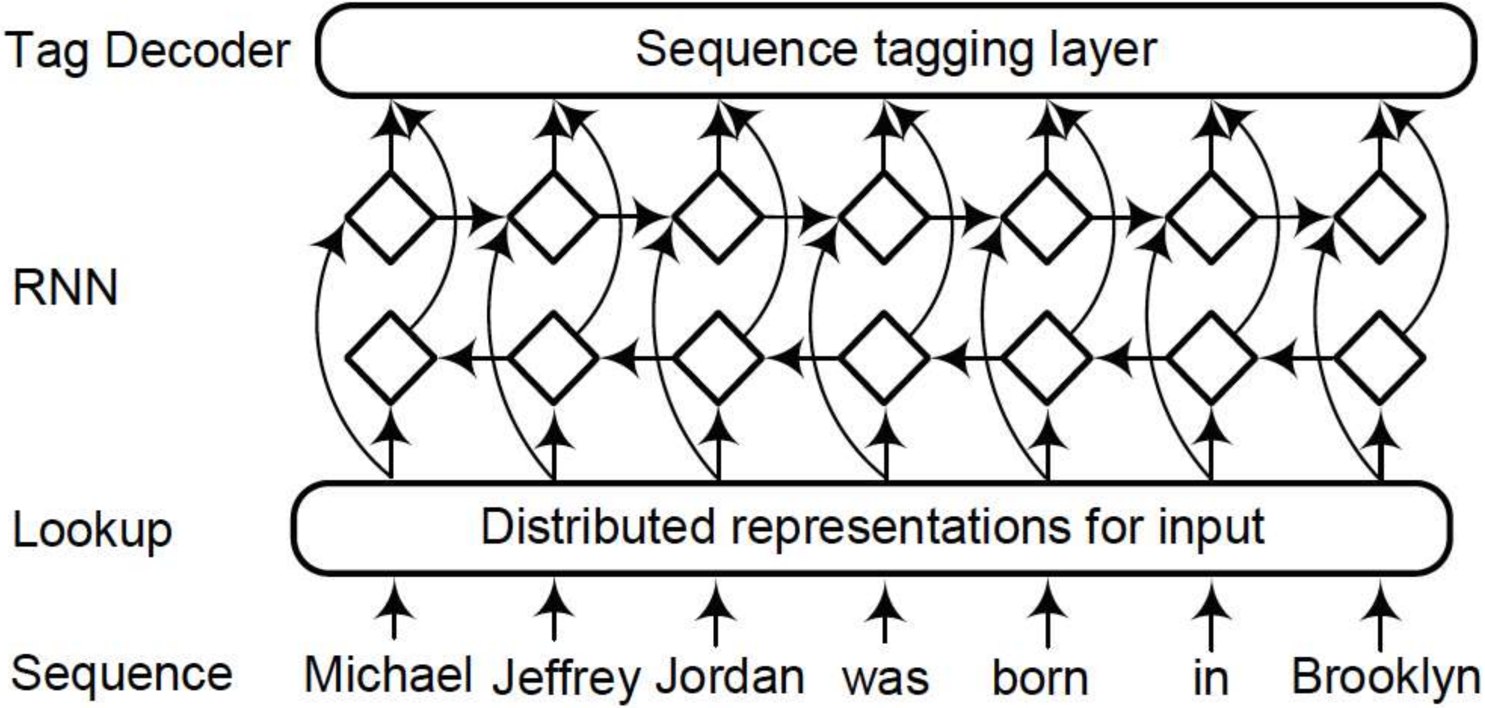
[Li, et. al. 2018]

Context Layer - CNN



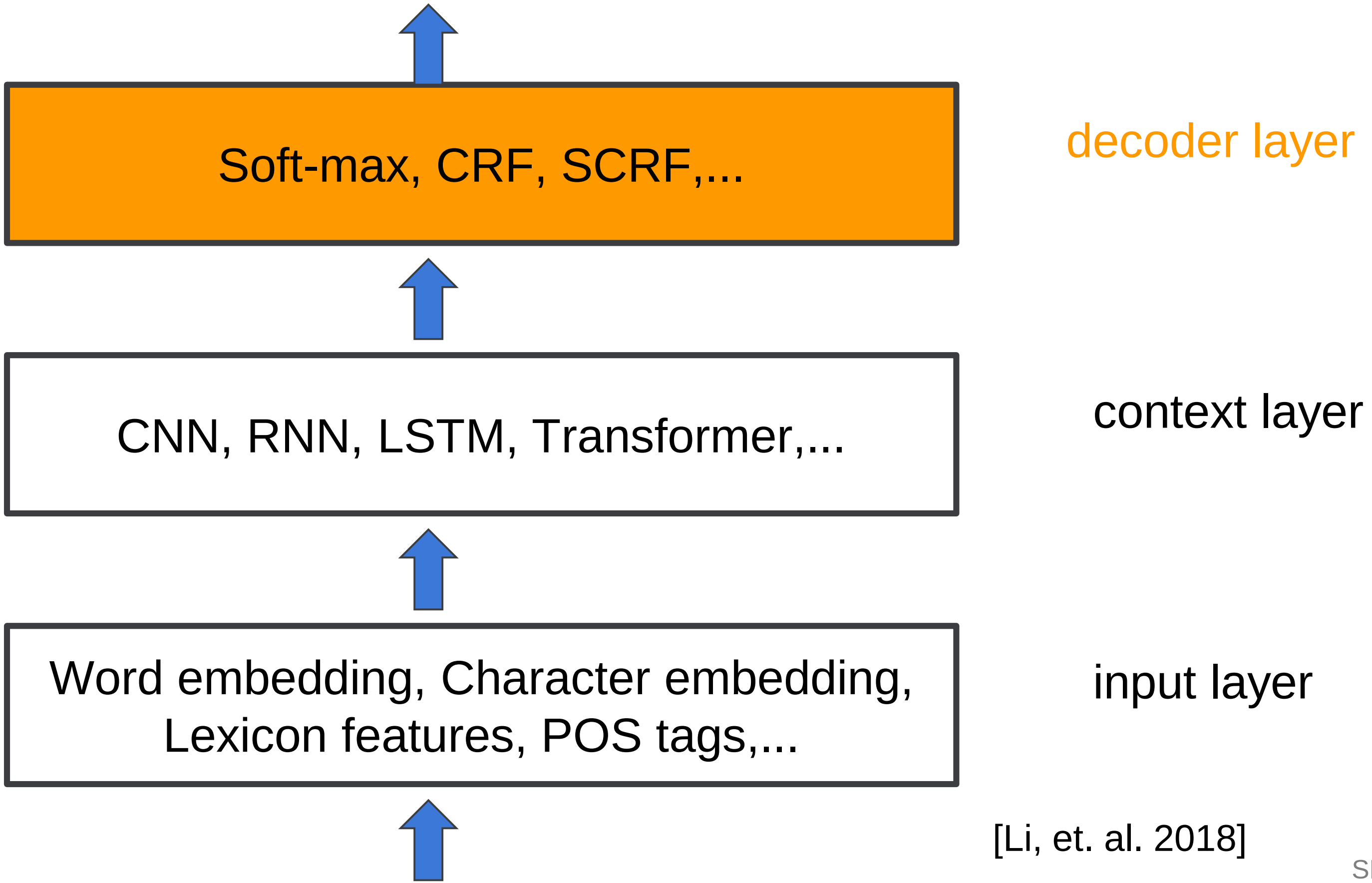
[Li et. al. 2018]

Context Layer - RNN



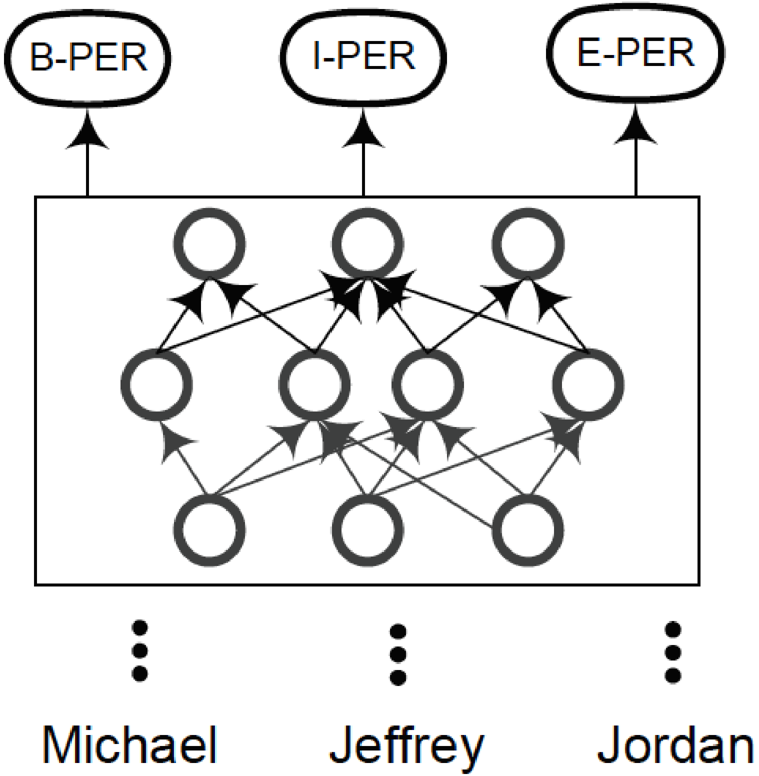
[Li et. al. 2018]

Deep Learning Tagger Architecture



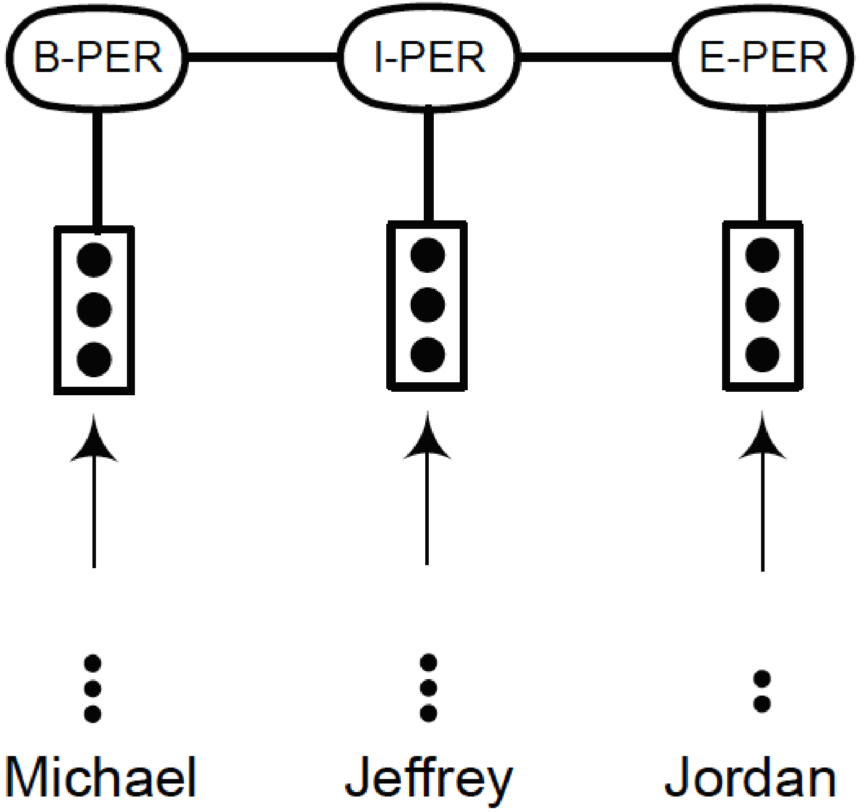
[Li, et. al. 2018]

Tag Decoder - MLP+softmax



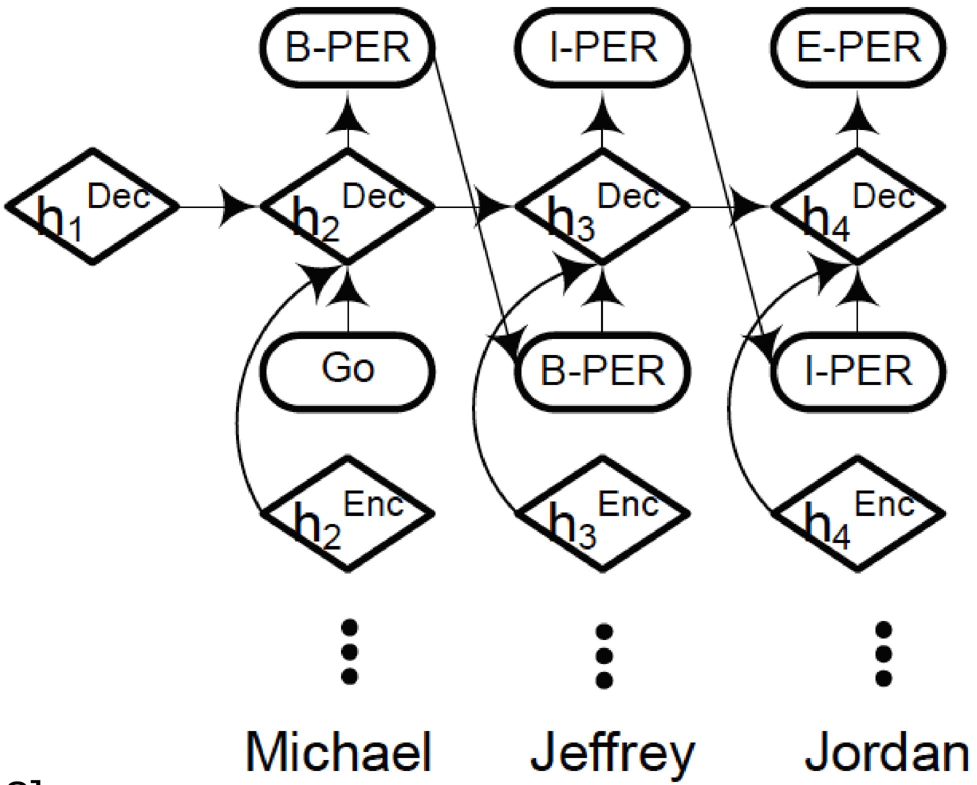
[Li et. al. 2018]

Tag Decoder - CRF



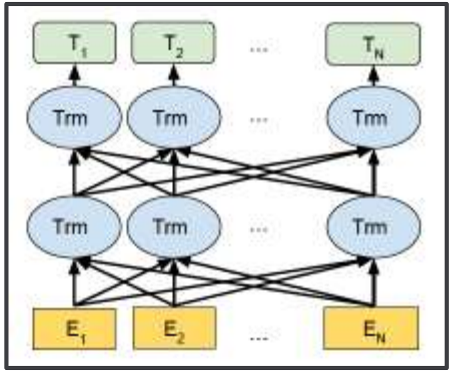
[Li et. al. 2018]

Tag Decoder - RNN



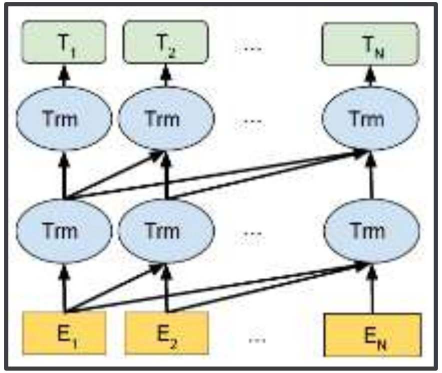
[Li et. al. 2018]

Pre-Training and Fine-Tuning



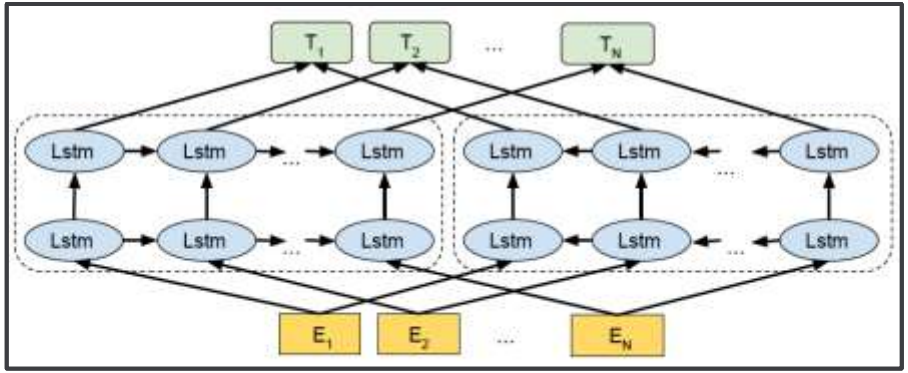
BERT

[Devlin et. al. 2018]



GPT

[Radford et. al. 2018]



ELMo

[Peters et. al. 2018]

Entity Tagging Evaluation

Exact-match evaluation

- segment match
- tag match

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1-score} = \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Entity Tagging Training Dataset

Corpus	Year	Text Source	#Tags	URL
MUC-6	1995	Wall Street Journal texts	7	https://catalog.ldc.upenn.edu/LDC2003T13
MUC-6 Plus	1995	Additional news to MUC-6	7	https://catalog.ldc.upenn.edu/LDC96T10
MUC-7	1997	New York Times news	7	https://catalog.ldc.upenn.edu/LDC2001T02
CoNLL03	2003	Reuters news	4	https://www.clips.uantwerpen.be/conll2003/ner/
ACE	2000 - 2008	Transcripts, news	7	https://www.ldc.upenn.edu/collaborations/past-projects/ace
OntoNotes	2007 - 2012	Magazine, news, conversation, web	89	https://catalog.ldc.upenn.edu/LDC2013T19
W-NUT	2015 - 2018	User-generated text	18	http://noisy-text.github.io
BBN	2005	Wall Street Journal texts	64	https://catalog.ldc.upenn.edu/ldc2005t33
NYT	2008	New York Times texts	5	https://catalog.ldc.upenn.edu/LDC2008T19
WikiGold	2009	Wikipedia	4	https://figshare.com/articles/Learning_multilingual_named_entity_recognition_from_Wikipedia/5462500
WiNER	2012	Wikipedia	4	http://rali.iro.umontreal.ca/rali/en/winer-wikipedia-for-ner
WikiFiger	2012	Wikipedia	113	https://github.com/xiaoling/figer
N^3	2014	News	3	http://aksw.org/Projects/N3NERNEDNIF.html
GENIA	2004	Biology and clinical texts	36	http://www.geniaproject.org/home
GENETAG	2005	MEDLINE	2	https://sourceforge.net/projects/bioc/files/
FSU-PRGE	2010	PubMed and MEDLINE	5	https://julielab.de/Resources/FSU_PRGE.html
NCBI-Disease	2014	PubMed	790	https://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/
BC5CDR	2015	PubMed	3	http://bioc.sourceforge.net/
DFKI	2018	Business news and social media	7	https://dfki-lt-re-group.bitbucket.io/product-corpus/

[Li et. al. 2018]

Entity Tagging on CoNLL03 English

Source	Method	F1 score
[Passos et al. 2014]	CRF	90.90
[Huang et al. 2015]	Bi-LSTM+CRF	84.26
[Collobert et al. 2011]	Conv-CRF	89.59
[Kuru et al. 2016]	char embedding	84.52
[Chiu and Nichols 2015]	word + char embedding	91.62
[Devlin et. al. 2018]	BERT Large	92.8

References - Entity Tagging

- [Baum et. al. 1966] Baum, L. E.; Petrie, T. (1966). "Statistical Inference for Probabilistic Functions of Finite State Markov Chains". The Annals of Mathematical Statistics. 37 (6): 1554–1563.
- [Chiu and Nichols, 2015] Jason PC Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional lstm-cnns. arXiv preprint arXiv:1511.08308
- [Collins and Singer 1999] Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.
- [Collobert et. al. 2011] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug):2493–2537.
- [Collobert and Weston 2008] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pages 160–167. ACM.
- [Devlin et. al. 2018] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805, 2018
- [Huang et. al. 2015] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991
- [Kuru et. al. 2016] Onur Kuru, Ozan Arkan Can, and Deniz Yuret. 2016. Charner: Character-level named entity recognition. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 911–921
- [Lafferty et. al. 2001] Lafferty, J., McCallum, A., Pereira, F. (2001). "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". Proc. 18th International Conf. on Machine Learning. Morgan Kaufmann. pp. 282–289
- [Li et. al. 2018] Jing Li, Aixin Sun, Jianglei Han, Chenliang Li, A Survey on Deep Learning for Named Entity Recognition, Dec. 2018, arXiv preprint, https://arxiv.org/abs/1812.09449
- [McCallum et. al. 2000] McCallum, Andrew; Freitag, Dayne; Pereira, Fernando (2000). "Maximum Entropy Markov Models for Information Extraction and Segmentation" (PDF). Proc. ICML 2000. pp. 591–598

References (continued)

- [Passos 2014] Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. arXiv preprint arXiv:1404.5367
- [Peters et. al. 2018] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In NAACL.
- [Radford et.al. 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.
- [Yadav et. al. 2018] Vikas Yadav, Steven Bethard, A Survey on Recent Advances in Named Entity Recognition from Deep Learning models, Proceedings of the 27th International Conference on Computational Linguistics, pages 2145–2158 Santa Fe, New Mexico, USA, August 20-26, 2018.

Deep NLP in Search Systems

- Language Understanding
  - Entity Tagging: word level prediction
  - **Entity Disambiguation: knowledge base entity prediction**
  - Intent Classification: sentence level prediction
- Document Retrieval and Ranking
  - Efficient Candidate Retrieval
  - Deep Ranking Models
- Language Generation for Search Assistance
  - Query Suggestion: word-level sequence to sequence
  - Spell Correction: character-level sequence to sequence
  - Auto Complete: partial sequence to sequence

Entity Disambiguation

- Problem statement
- Motivation
- Challenges
- System architecture
  - Candidate Entity Generation
  - Candidate Entity Selection
  - Joint entity tagging and disambiguation
- Evaluation

Entity Disambiguation - Problem Statement

- Also known as entity linking
- Resolves mentions to entities in a given knowledge base
  - Mentions are mostly named entities
  - Example of knowledge base: freebase, wikipedia
- Examples: Jaguar
  - The prey saw the jaguar cross the jungle. *KB:Jaguar*
  - The man saw a <u>jaguar</u> speed on the highway. *KB:Jaguar_Cars*

Motivation

Increase the quality of the retrieved results

Example:

Which Michael Jordan were you looking for?

Michael Jordan (disambiguation)

From Wikipedia, the free encyclopedia

Michael or **Mike Jordan** may refer to:

People [edit]

Sports [edit]

- Michael Jordan (born 1963), American basketball player and businessman
- Michael Jordan (footballer) (born 1986), English goalkeeper
- Mike Jordan (racing driver) (born 1958), English racing driver
- Mike Jordan (baseball, born 1863) (1863–1940), baseball player
- Mike Jordan (cornerback) (born 1992), American football cornerback
- Michael Jordan (offensive lineman), American football offensive lineman
- Michael-Hakim Jordan (born 1977), American professional basketball player
- Michal Jordán (born 1990), Czech ice hockey player

Other people [edit]

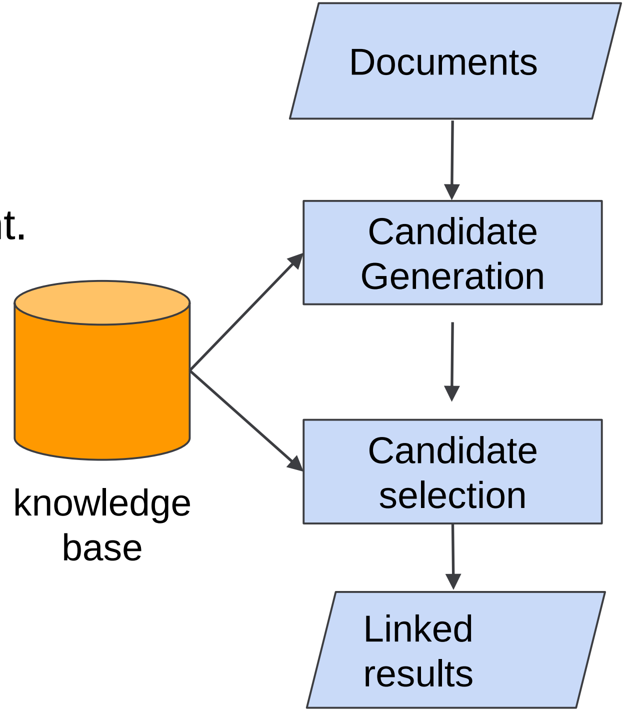
- Michael B. Jordan (born 1987), American actor
- Michael I. Jordan (born 1956), American researcher in machine learning and artificial intelligence
- Michael Jordan (insolvency baron) (born 1931), English businessman
- Michael Jordan (Irish politician), Irish Farmers' Party TD from Wexford, 1927–1932
- Michael H. Jordan (1936–2010), American executive for CBS, PepsiCo, Westinghouse
- Michael Jordan (mycologist), English mycologist

Challenges

- Name variation: *New York* vs *Big Apple*
- Ambiguity: *Michael Jordan*
- Metonymy: *Beijing* (city or Chinese government)
- Absence: no entries in knowledge base
- Evolving information: new company names, e.g. *tic-tok*.

System Architecture

1. Candidate entity generation
   a. Name dictionary based techniques.
   b. Surface form expansion from the local document.
   c. Methods based on search engines.
2. Candidate entity selection
   a. Graph-based methods
   b. Text-based methods



Candidate Entity Generation

- Name dictionary based techniques
  - Entity pages
  - Redirect pages
  - Disambiguation pages
  - Bold phrases from the first paragraphs
  - Hyperlinks in Wikipedia articles
- Surface form expansion
  - Heuristic based methods
  - Supervised learning methods
- Methods based on search engines

[Shen et. al. 2014]

Michael Jordan (disambiguation)

From Wikipedia, the free encyclopedia

Michael or **Mike Jordan** may refer to:

People [edit]

Sports [edit]

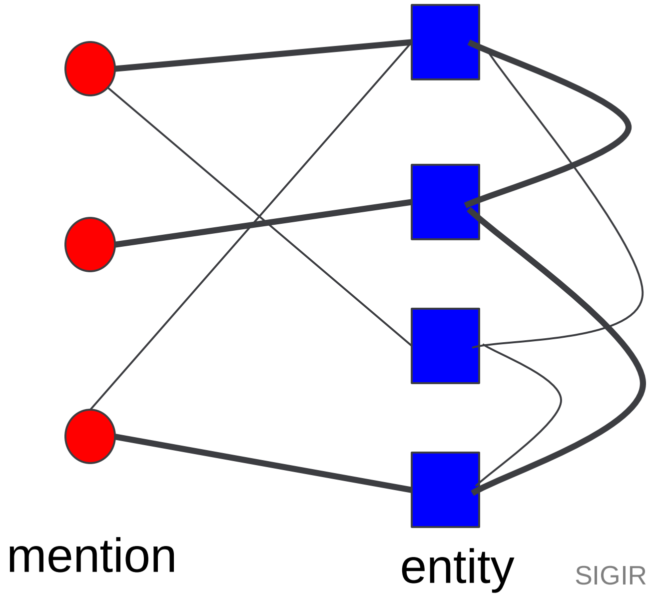
- Michael Jordan (born 1963), American basketball player and businessman
- Michael Jordan (footballer) (born 1986), English goalkeeper
- Mike Jordan (racing driver) (born 1958), English racing driver
- Mike Jordan (baseball, born 1863) (1863–1940), baseball player
- Mike Jordan (cornerback) (born 1992), American football cornerback
- Michael Jordan (offensive lineman), American football offensive lineman
- Michael-Hakim Jordan (born 1977), American professional basketball player
- Michal Jordán (born 1990), Czech ice hockey player

Other people [edit]

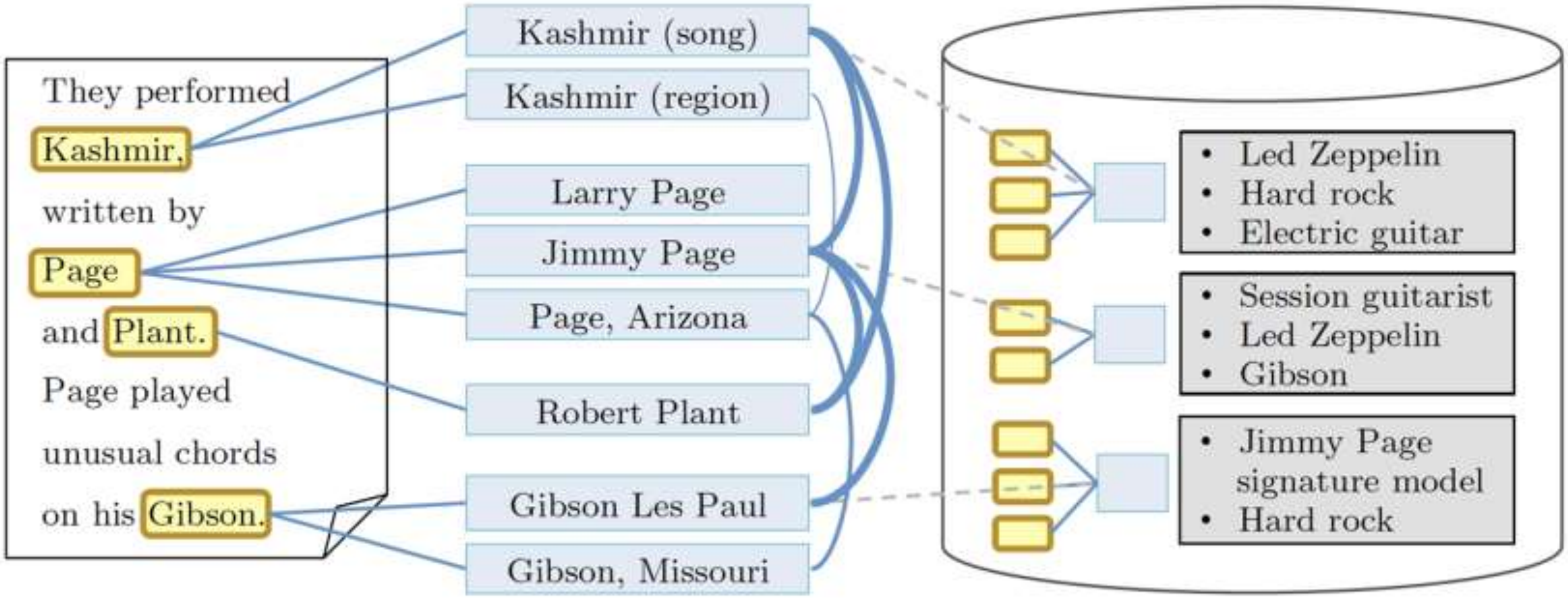
- Michael B. Jordan (born 1987), American actor
- Michael I. Jordan (born 1956), American researcher in machine learning and artificial intelligence
- Michael Jordan (insolvency baron) (born 1931), English businessman
- Michael Jordan (Irish politician), Irish Farmers' Party TD from Wexford, 1927–1932
- Michael H. Jordan (1936–2010), American executive for CBS, PepsiCo, Westinghouse
- Michael Jordan (mycologist), English mycologist

Candidate Entity Selection - Graph based models

- knowledge graph lends itself to graph based methods.
- Build a weighted undirected graph with mentions and candidate entities as nodes
  - weights on mention-entity edges.
  - weights on entity-entity edge.
- Compute an optimal subgraph:
  - optimality is model dependent.
  - contains all mentions
  - one entity per mention



An Example Graph



[Hoffart, et. al. 2011]

Candidate Entity Selection - Text-Based Models

- Idea: find agreement (similarity) between entity and mention.
  - Example: computational intelligence by *Michael Jordan* and Stuart Russell from UC Berkeley

- wiki/Michael_Jordan
- wiki/Michael_I._Jordan

- Models according to context.
  - Direct Models.
    - use mention and candidate entities only.
  - Local Models
    - use local context around the mention.
  - Coherence Models.
    - use mentions and entities in a document.
  - Collaborative Models.
    - use mentions and entities across related documents.

increasing context

Individual vs Joint Approach

Two types of approaches according to how mentions are resolved.

- Individual approach
  - Handle one mention at a time.
  - Rank the candidate entities.
  - Relatively low complexity.
- Joint approach
  - Treat all mentions in a document as a sequence.
  - Jointly optimize the mention-entity pairs.
  - High complexity, usually resorting to approximations.

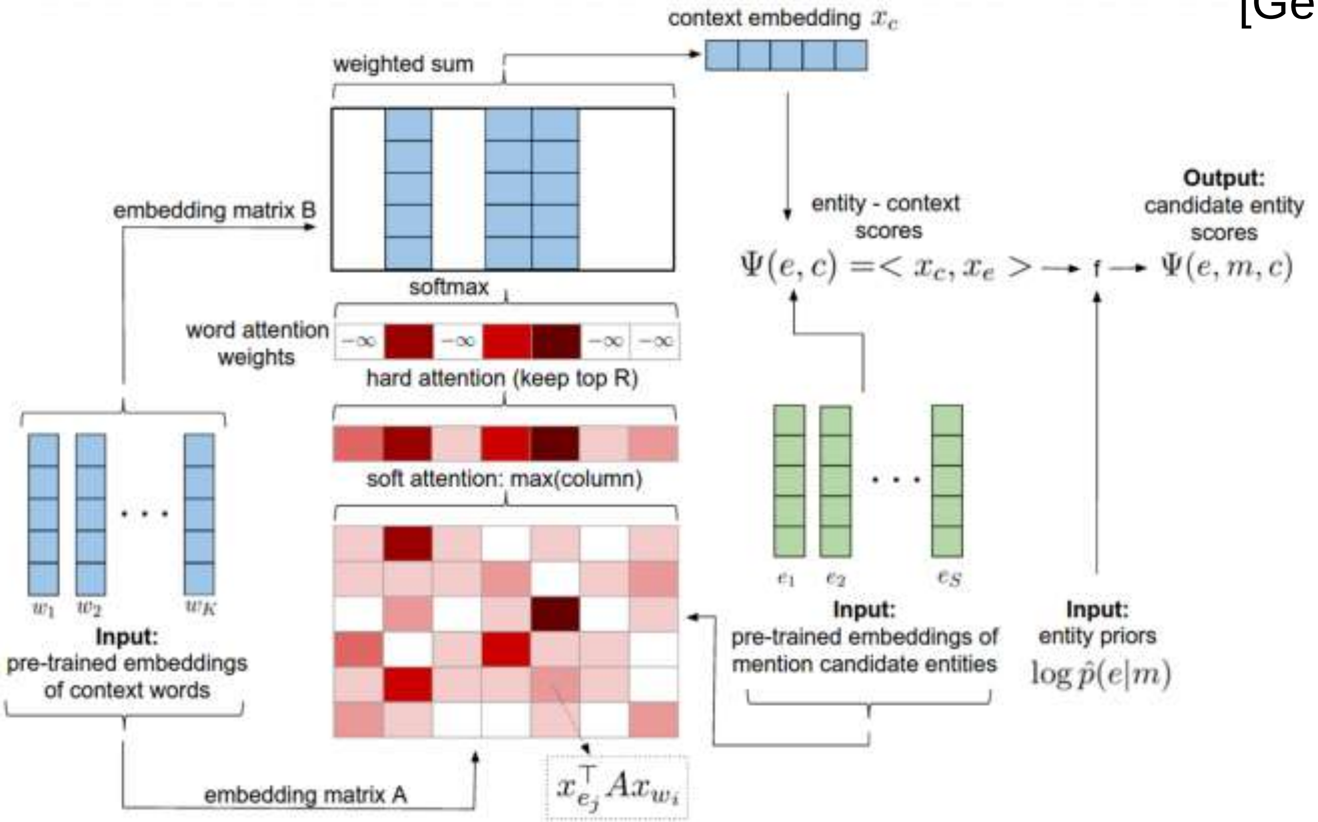
Features

Different features are used in candidate selection modeling.

- Traditional Features
  - entity popularity, entity type.
  - surface form similarity between entity and mention.
  - coherence between mapped entities
- Deep Features [Yamada et. al. 2016]
  - word embedding
  - entity embedding
  - similarity between words, context and candidate entities.

Example: Entity Disambiguation with Attention

[Genea et. al. 2017]



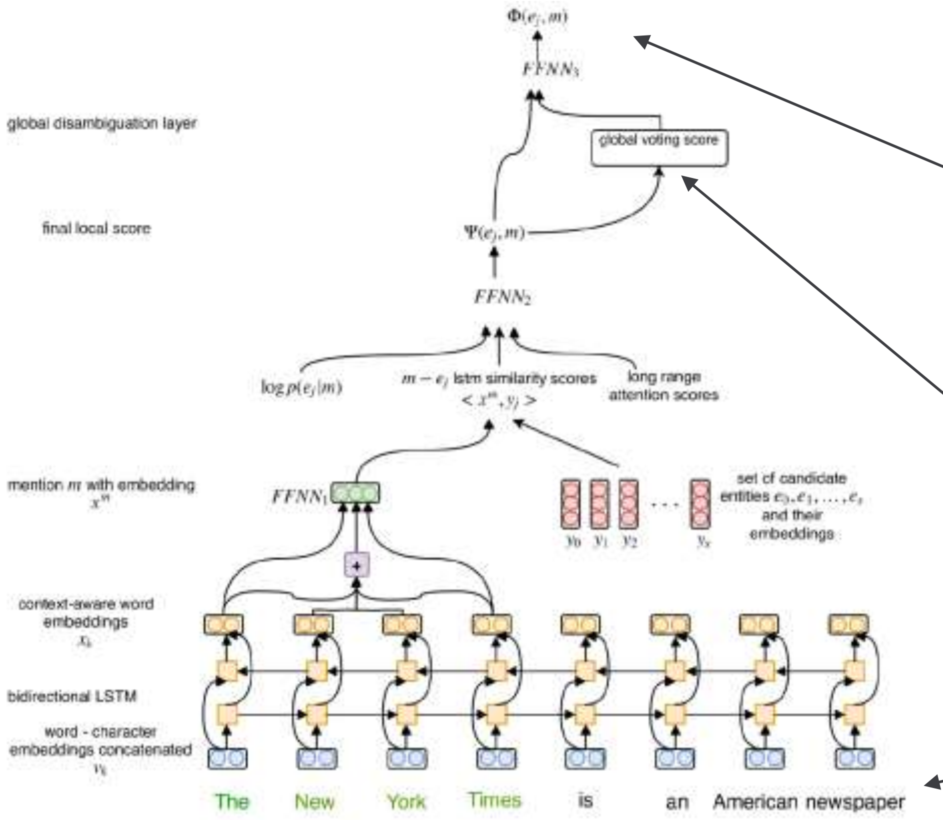
training:
max-margin loss

Joint Entity Tagging and Disambiguation

- Previously, we find entities first, then link them. But if entities are identified wrong, disambiguation will likely fail too.
  - Over-split: <u>Romeo</u> and <u>Juliet</u> by Shakespeare
  - Under-split: Baby <u>Romeo and Juliet</u> were born hours apart.
- We could do entity tagging and disambiguation jointly. [Sil, et. al.2013]
  - Over-generate candidate mentions
  - Generate possible entities per mention
  - Score non-overlapping mention-entity pair jointly.

A Joint Neural Entity Tagging and Disambiguation

[Kolitsas, et. al. 2018]



- Trained on golden mention-entity pairs.
- Used max-margin loss.

cosine similarity between current entity and average of other entities in the document

Select mentions that have at least one possible entity

Entity Disambiguation Evaluation

- Entity-tagging-style F1 score
  - A link is considered correct only if the mention matches the gold boundary and the linked entity is also correct
- Accuracy
  - another common metric, simpler than F1 score.
- TAC-KBP B-Cubed+ F1 score
  - not widely used

References - Named Entity Disambiguation

- [Cucerzan 2007] Cucerzan, Silviu. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL): 708–716.
- [Ganea et. al. 2017] Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2609–2619. Association for Computational Linguistics.
- [Globerson et. al. 2016] Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2016. Collective entity resolution with multi-focal attention. In ACL (1).
- [Han et. al. 2011] Han, Xianpei; Sun, Le; Zhao, Jun. Collective Entity Linking in Web Text: A Graph-based Method.
- [Hoffart, et. al. 2011] Johannes Hoffart, Mohamed A. Yosef, Ilaria Bordino, Hagen F¨urstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In Proceedings of the Conference on Empirical Method in Natural Language Processing, pages 782–792. Association for Computational Linguistics.
- [Kolitsas et al., 2018] Nikolaos Kolitsas, Octavianeugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. In CoNLL, 2018.
- [Kulkarni et. al. 2009] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, Collective annotation of Wikipedia entities in web text, in SIGKDD, 2009, pp. 457–466
- [Ling et al.2015] Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design Challenges for Entity Linking. Transactions of the Association for Computational Linguistics, 3:315–328.
- [Le et. al., 2018] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. In ACL, 2018.
- [Rao et. al. 2013] Rao, Delip; McNamee, Paul; Dredze, Mark (2013). Entity Linking: Finding Extracted Entities in a Knowledge Base. Multi-source, Multilingual Information Extraction and Summarization. Springer Berlin Heidelberg: 93–115

References (continued)

- [Raiman et. al. 2018] Jonathan Raiman and Olivier Raiman. 2018. DeepType: Multilingual Entity Linking by Neural Type System Evolution. In Proc. of AAAI.
- [Shen et. al. 2014] Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. TKDE
- [Sil et. al. 2013] Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pages 2369–2374. ACM
- [Sil et. al. 2016] Sil, A., and Florian, R. 2016. One for all: Towards language independent named entity linking. ACL.
- [Yamada et. al. 2016] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. CoNLL 2016, page 250.

Deep NLP in Search Systems

- Language Understanding
  - Entity Tagging: word level prediction
  - Entity Disambiguation: knowledge base entity prediction
  - **Intent Classification: sentence level prediction**
- Document Retrieval and Ranking
  - Efficient Candidate Retrieval
  - Deep Ranking Models
- Language Generation for Search Assistance
  - Query Suggestion: word-level sequence to sequence
  - Spell Correction: character-level sequence to sequence
  - Auto Complete: partial sequence to sequence

Intent Classification

- Problem statement
- Deep Learning Models
  - fastText
  - CNN
  - Bi-RNN + Attention

Intent Classification - Problem Statement

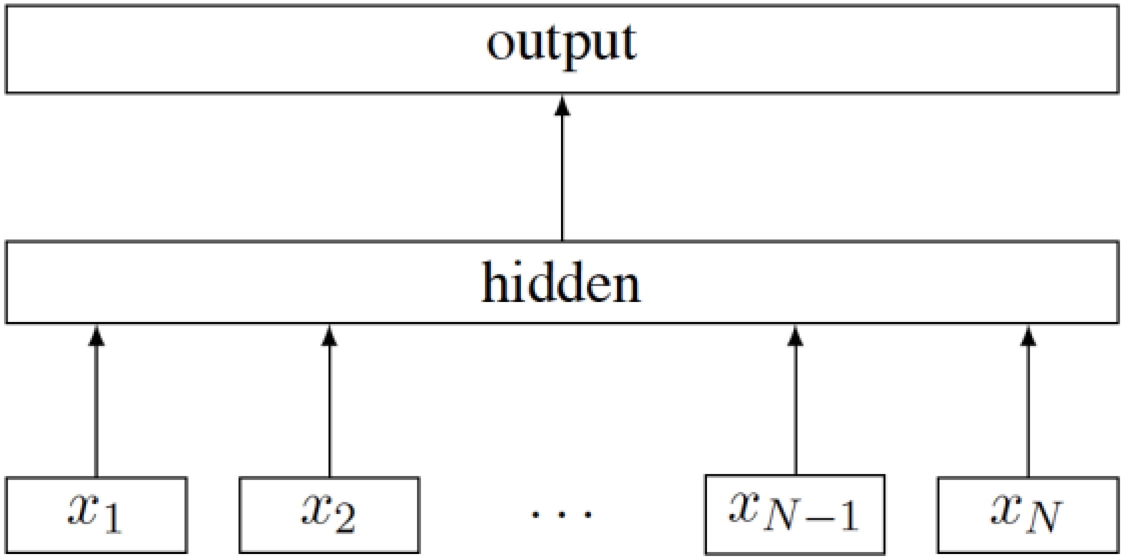
- Web search intent can be classified into 3 class: [Broder 2002]
  - **Navigational**. The immediate intent is to reach a particular site.
    - Greyhound Bus. Probable target http://www.greyhound.com
  - **Informational**. The intent is to acquire some information assumed to be present on one or more web pages.
    - San Francisco
  - **Transactional**. The intent is to perform some web-mediated activity.
    - Shopping activities
- In conversational AI, intent is usually task-specific, e.g.
  - I would like to book a flight from SFO to CDG: **FlightBooking**
  - What software can I use to view epub documents: **SoftwareRecommendation**.
- We focus on the latter in this tutorial.

Intent Classification - Methods

- Traditional methods
  - Features: bag of words, n-gram, TF-IDF.
  - Models: logistic regression, naive Bayes, SVM, random forest.
- Deep learning methods
  - Word embedding + linear classifier (fastText) [Joulin et. al. 2016].
  - Convolutional neural networks [Hashemi 2016] .
  - Bi-RNN + attention (joint slot filling) [Liu et al.2016] .

Intent Classification - fastText

[Joulin et. al. 2016]



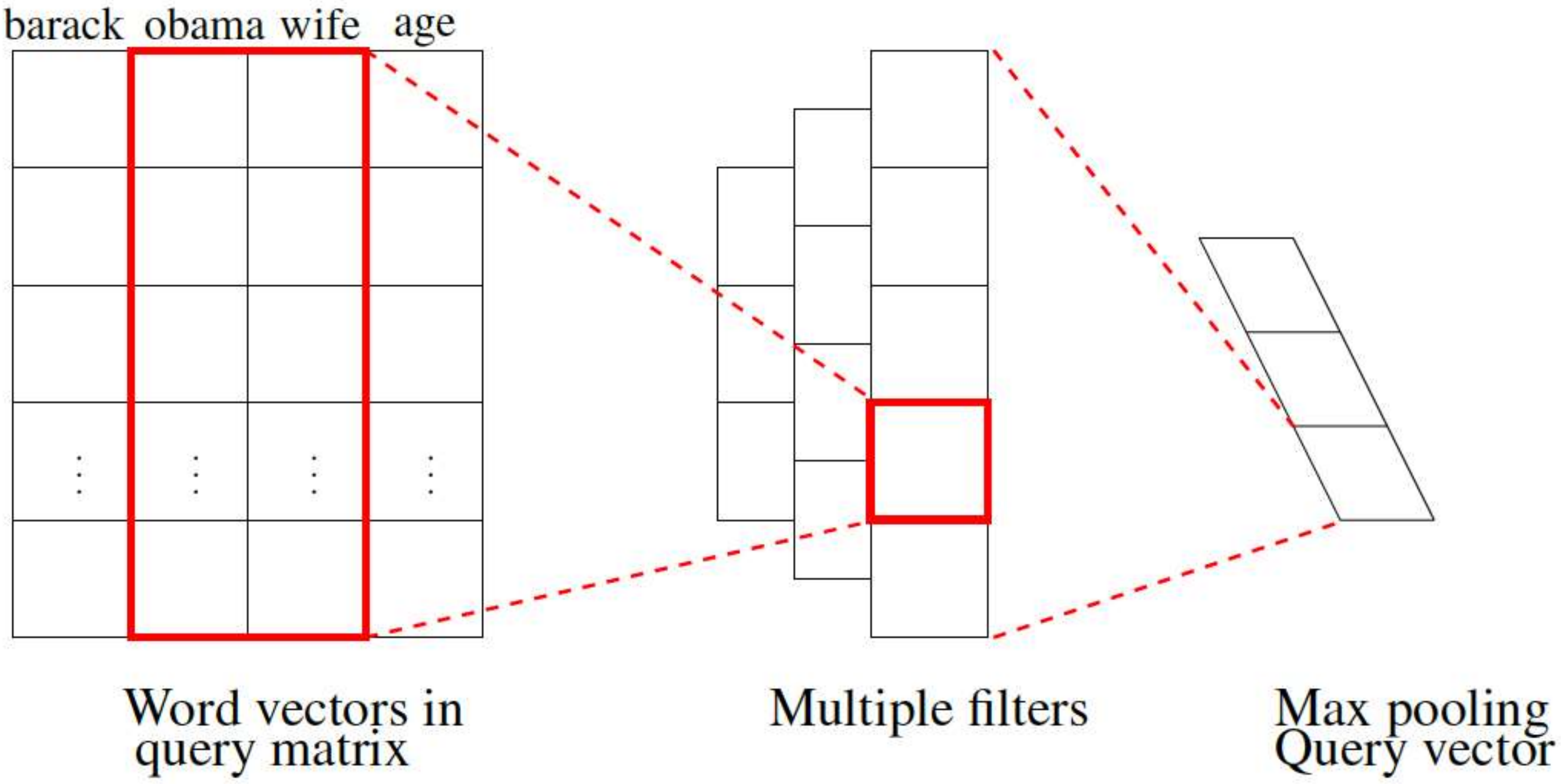
softmax

Average

word and N gram embedding features

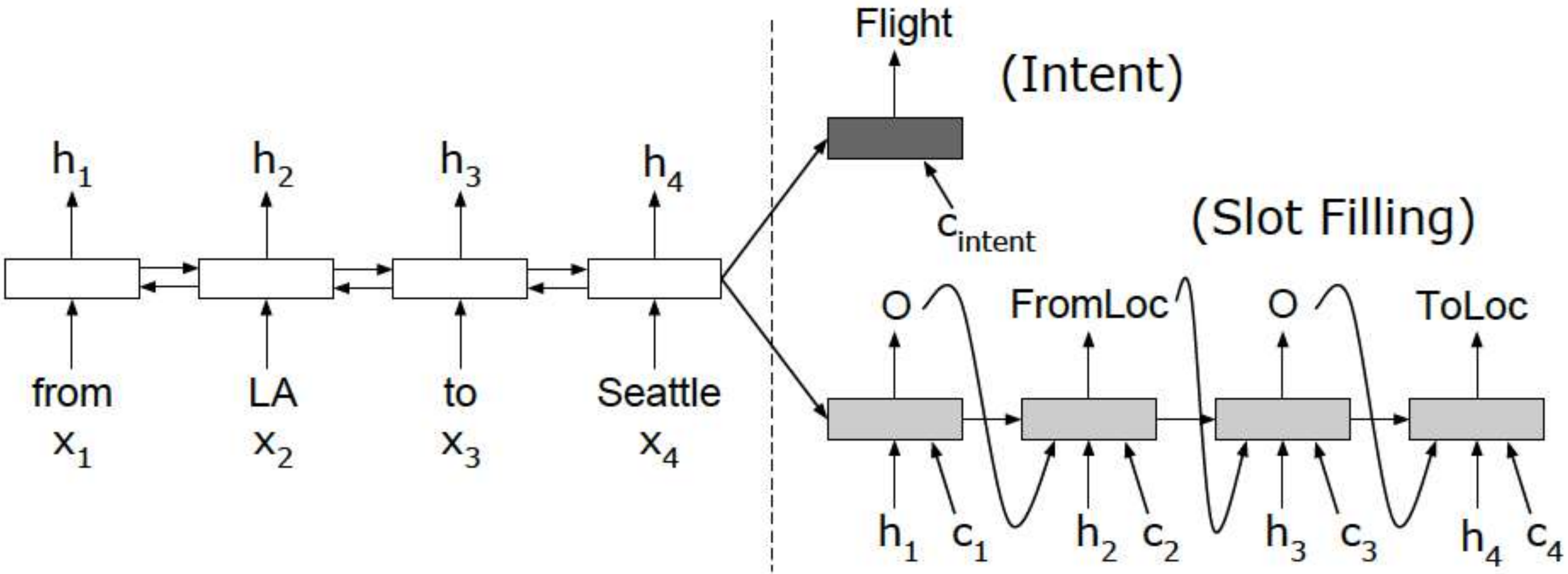
Intent Classification - CNN

[Hashemi 2016]



Intent Classification - Bi-RNN+Attention

[Liu et. al. 2016]



References - Intention Classification

- [Broder 2002] A. Broder. A taxonomy of web search. SIGIR Forum, 36(2):3–10, 2002.
- [Kim 2014] Y. Kim. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.
- [Joulin et. al. 2016] Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL).
- [Lai et al.2015] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In AAAI, pages 2267–2273
- [Liu et al.2016] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101
- [Zhou et al.2016] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In The 54th Annual Meeting of the Association for Computational Linguistics, page 207.
- [Kim et. al. 2016] Joo-Kyung Kim, Gokhan Tur, Asli Celikyilmaz, Bin Cao, and Ye-Yi Wang. 2016. Intent detection using semantically enriched word embeddings. In Proceedings of SLT.
- [Hashemi 2016] Homa B Hashemi, Amir Asiaee, and Reiner Kraft. 2016. Query intent detection using convolutional neural networks. In International Conference on Web Search and Data Mining, Workshop on Query Understanding.
- [Liu et. al. 2016]. Liu and I. Lane, “Attention-based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling,” in Interspeech, 2016.
- [Shi et. al. 2016] Y. Shi, K. Yao, L. Tian, and D. Jiang, “Deep lstm based feature mapping for query classification,” in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1501–1511.

Deep NLP in Search Systems

- Document Retrieval & Ranking

Jun Shi, Weiwei Guo

Deep NLP in Search Systems

- Language Understanding
  - Entity Tagging: word level prediction
  - Entity Disambiguation: knowledge base entity prediction
  - Intent Classification: sentence level prediction
- **Document Retrieval and Ranking**
  - **Efficient Candidate Retrieval**
  - **Deep Ranking Models**
- Language Generation for Search Assistance
  - Query Suggestion: word-level sequence to sequence
  - Spell Correction: character-level sequence to sequence
  - Auto Complete: partial sequence to sequence

Document Retrieval and Ranking

- Efficient Candidate Retrieval
- Deep Neural Ranking

Efficient Candidate Retrieval

- Syntactic retrieval
  - based on string matching
  - use inverted index
  - can include different fields (name, title, etc).
- Semantic retrieval
  - based on vector space representation.
  - approximate nearest neighbor search

Syntactic Retrieval

Query: mike software engineer

Inverted index

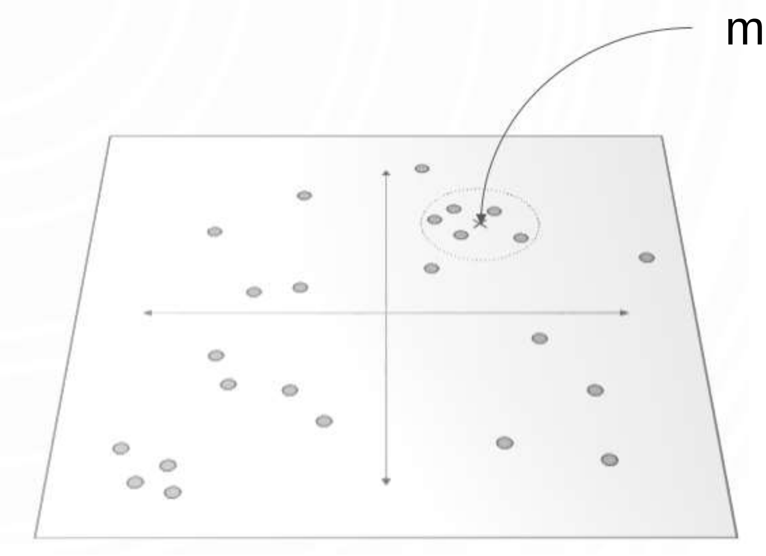
mike	Doc1, **Doc2**, **Doc4**, ...
software	**Doc2**, Doc3, **Doc4**, Doc7, ...
engineer	Doc1, **Doc2**, **Doc4**, Doc9, ...

Results: **Doc2, Doc4**

It won’t retrieve a document contains “mike is a software developer,...”

Semantic Retrieval - Concept

mike software engineer $\xrightarrow{\text{embedding}}$ [0.3, 1.27, -3.4, ...]



mike software engineer

The problem becomes nearest neighbor search

[Mitra et. al. 2017]

Semantic Retrieval - Vector Generation

- Bag of words
  - count based, TF-IDF vectors.
- Embedding vectors
  - word embedding: word2vec, Glove, BERT, etc.
  - sentence/document embedding
    - universal sentence encoder [Cer et. al. 2018]
    - Gaussian document representation [Giannis et. al. 2017]
    - power mean concatenation [Rücklé et. al. 2018]

Approximate Nearest Neighbor Search

- Exact nearest neighbor search
  - complexity is linear with the size of dataset, not suitable for large dataset.
- Approximate nearest neighbor search
  - allow bounded errors.
  - complexity is sublinear
  - many algorithms available
    - product quantization [Jégou et. al. 2016]
    - hierarchical navigable small world graphs [Malkov et. al. 2016]

References - Efficient Candidate Retrieval

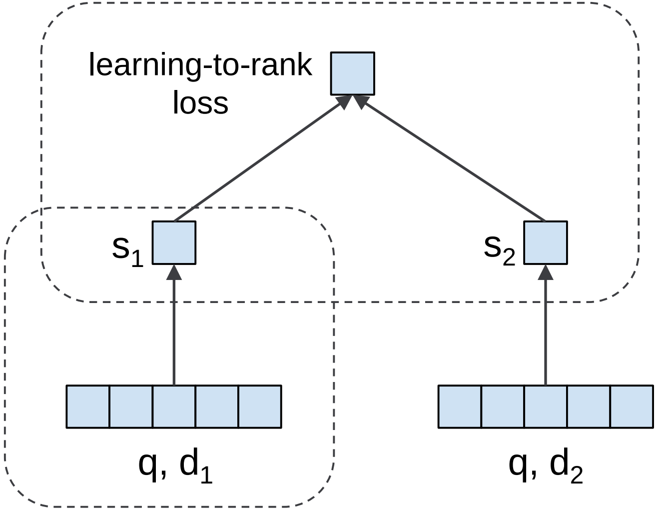
- [Cer et. al. 2018] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder. CoRR, abs/1803.11175, 2018. URL http://arxiv.org/abs/1803.11175.
- [Giannis et. al. 2017] Giannis Nikolentzos, Polykarpos Meladianos, Franc¸ois Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. 2017. Multivariate gaussian document representation from word embeddings for text categorization. In EACL
- [Manning et. al. 2008] Christopher D. Manning Prabhakar Raghavan Hinrich Schütze, Introduction to Information Retrieval Cambridge University Press New York, NY, USA, 2008
- [Mitra et. al. 2017] Mitra, B., & Craswell, N. (2017). Neural models for Information Retrieval. CoRR, abs/1705.01509.
- [Rücklé et. al. 2018] A. Rücklé, S. Eger, M. Peyrard, and I. Gurevych. Concatenated p-mean word embeddings as universal cross-lingual sentence representations. CoRR, abs/1803.01400, 2018. URL http://arxiv.org/abs/1803.01400.
- [Malkov et. al. 2016] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. arXiv:1603.09320, 2016
- [Jégou et. al. 2016] Hervé Jégou, Matthijs Douze, Cordelia Schmid. Product Quantization for Nearest Neighbor Search. IEEE Transactions on Pattern Analysis and Machine Intelligence, Institute of Electrical and Electronics Engineers, 2011, 33 (1), pp.117-128. ff10.1109/TPAMI.2010.57ff. ffinria-00514462v2

Deep NLP in Search Systems

- Language Understanding
  - Entity Tagging: word level prediction
  - Entity Disambiguation: knowledge base entity prediction
  - Intent Classification: sentence level prediction
- Document Retrieval and Ranking
  - Efficient Candidate Retrieval
  - **Deep Ranking Models**
- Language Generation for Search Assistance
  - Auto Completion
  - Query Reformulation
  - Spell Correction

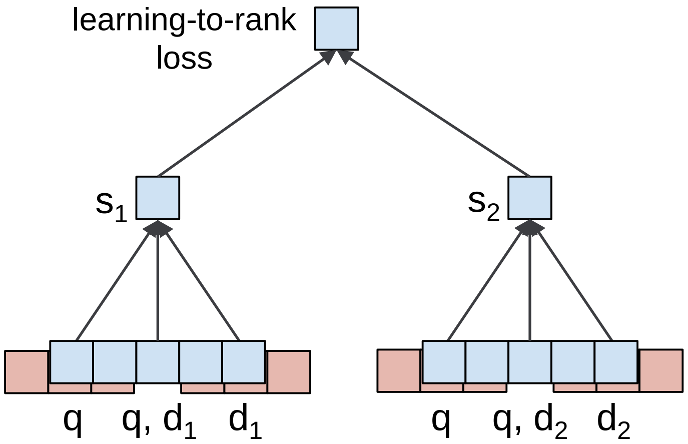
Deep Neural Ranking - Agenda

- Traditional methods
  - Ranking features
  - Learning to rank



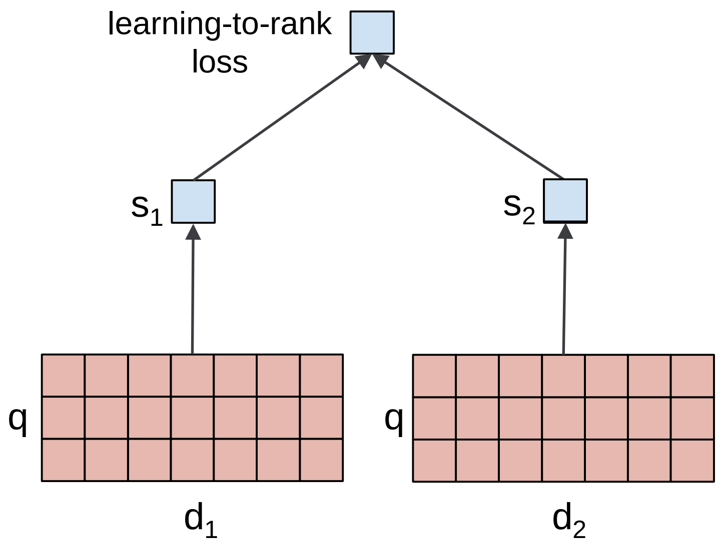
Deep Neural Ranking - Agenda

- Traditional methods
  - Ranking features
  - Learning to rank
- Deep neural ranking
  - Siamese Networks



Deep Neural Ranking - Agenda

- Traditional methods
  - Ranking features
  - Learning to rank
- Deep neural ranking
  - Siamese Networks
  - Interaction-based Networks



Traditional Ranking Features

- Hand-crafted features
  - query/document matching features
    - Cosine similarity between query and doc title
    - Clickthrough rate from query to this doc based on search log
    - …….
  - Document alone
    - popularity
    - number of incoming links
    - ……

Learning to Rank

(Burges, 2010)

- Pointwise ranking
  - Logistic regression

$$\frac{1}{1+e^{-s}} \quad \text{for } y = 1$$

- Pairwise ranking

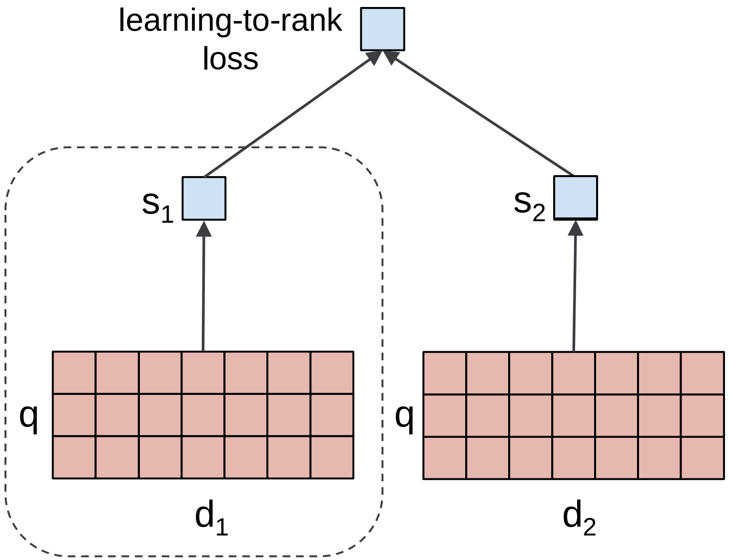
$$\frac{1}{1+e^{-(s_1-s_2)}} = \frac{e^{s_1}}{e^{s_1}+e^{s_2}}$$

- Listwise ranking
  - Cross entropy

$$\sum_i y_i \cdot \frac{e^{s_i}}{e^{s_1}+e^{s_2}+\ldots+e^{s_n}}$$

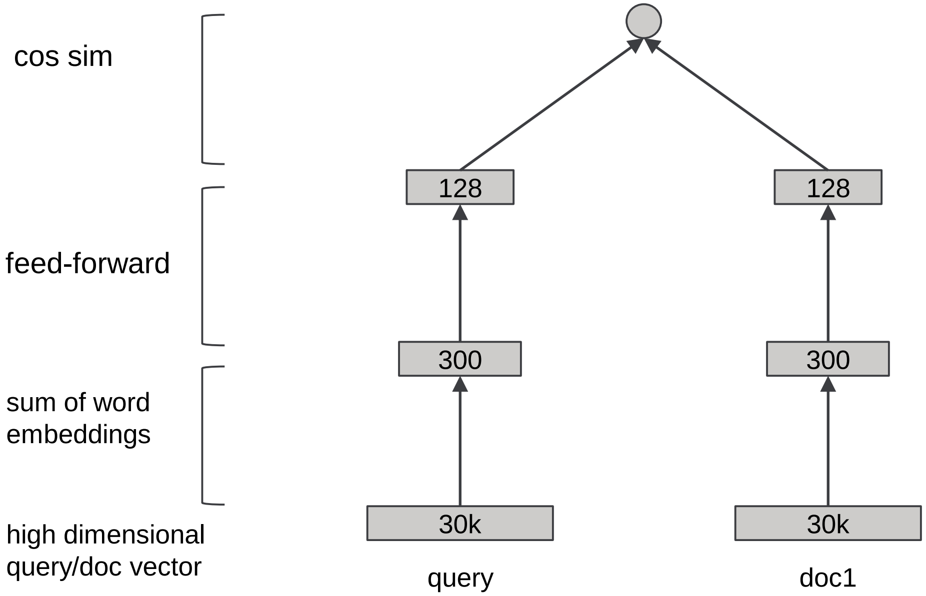
Deep Neural Ranking

- Focus on compute query/document score
- Two categories:
  - Siamese Networks
  - Interaction based Networks



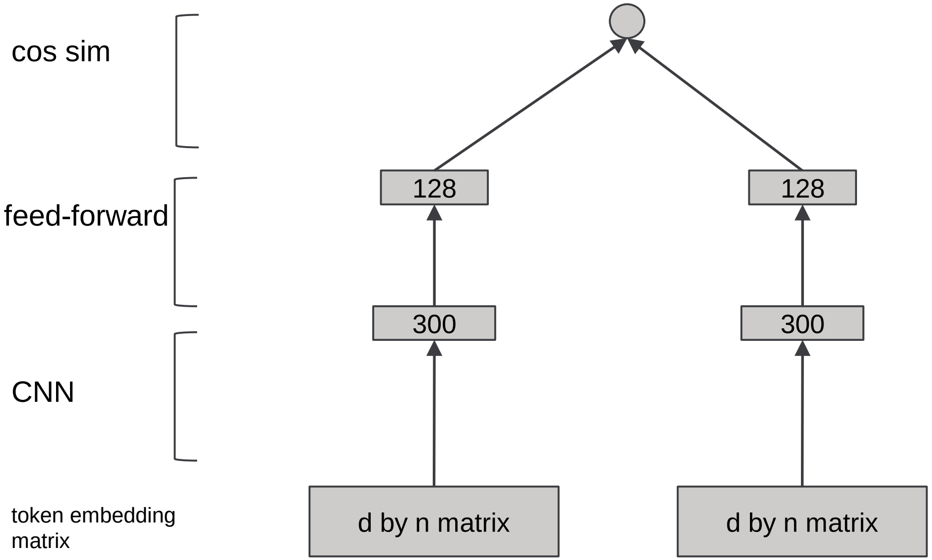
Deep Structured Semantic Model

(Huang et al., 2013)



Modeling Word Sequence by CNN

(Shen et al., 2014)

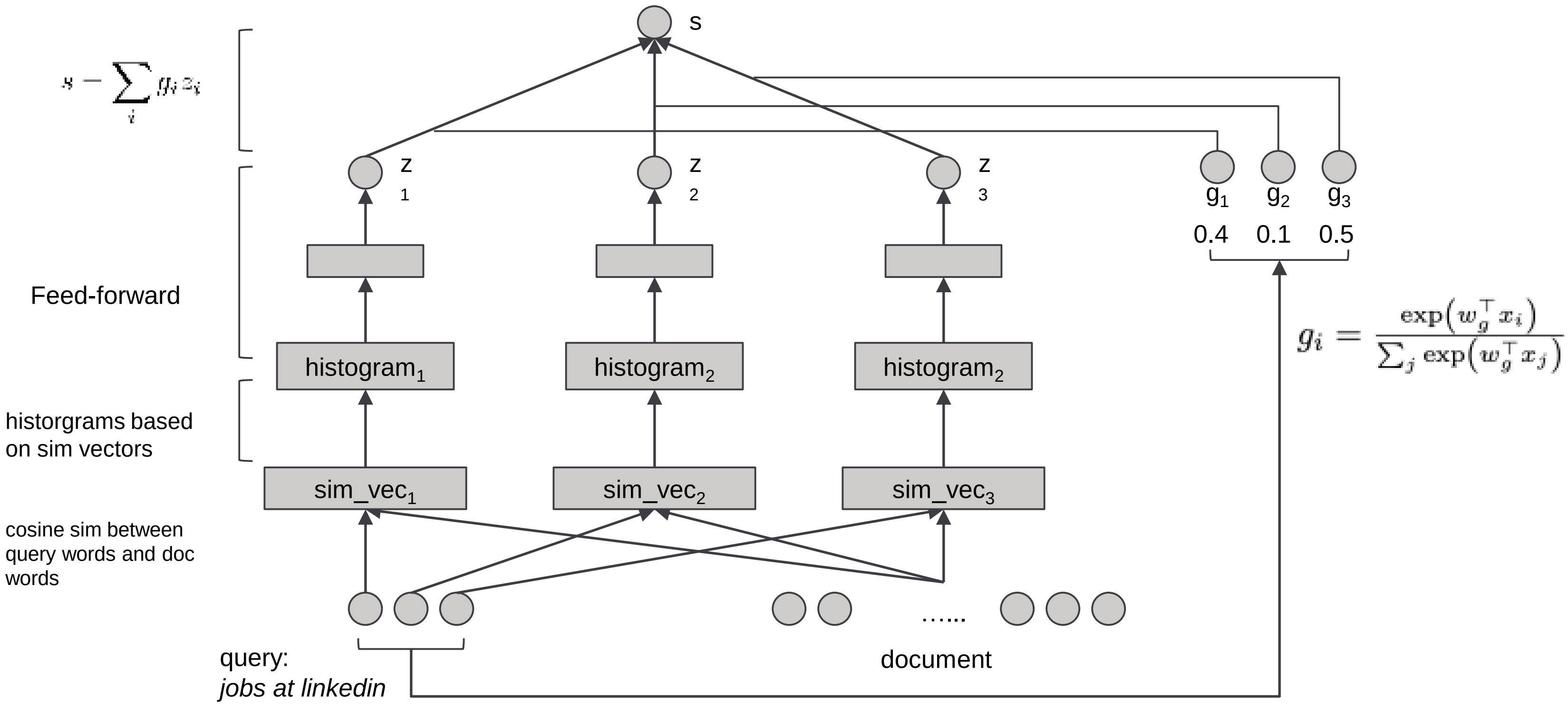


Siamese Networks

- Pros:
  - Generalization; semantic matching
  - Efficient; doc embs can be precomputed
- Cons:
  - Lexical features lost: people/company names, rare words

Interaction-based Networks

(Guo et al., 2016)



Deep Neural Ranking Summary

- Only focus on query/doc scoring
- End-to-end models
- Two popular architectures

	Siamese Network	**Interaction Network**
Match	topical matches	lexical matches
Latency	small	large

References - Deep Neural Ranking

- Huang, Po-Sen, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. "Learning deep structured semantic models for web search using clickthrough data." In Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pp. 2333-2338. ACM, 2013.
- Shen, Yelong, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. "A latent semantic model with convolutional-pooling structure for information retrieval." In Proceedings of the 23rd ACM international conference on conference on information and knowledge management, pp. 101-110. ACM, 2014.
- Rodrigo Nogueira, Kyunghyun Cho, Passage re-ranking with BERT, 2019
- Burges. "From ranknet to lambdarank to lambdamart: An overview." *Learning.* 2010.
- Guo, Jiafeng, Yixing Fan, Qingyao Ai, and W. Bruce Croft. "A deep relevance matching model for ad-hoc retrieval." In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 55-64. ACM, 2016.
- Mitra, Bhaskar, Fernando Diaz, and Nick Craswell. "Learning to match using local and distributed representations of text for web search." In Proceedings of the 26th International Conference on World Wide Web, pp. 1291-1299. International World Wide Web Conferences Steering Committee, 2017.

Deep NLP in Search Systems - Language Generation for Search Assistance

Weiwei Guo

Deep NLP in Search Systems

- Language Understanding
  - Entity Tagging: word level prediction
  - Entity Disambiguation: knowledge base entity prediction
  - Intent Classification: sentence level prediction
- Document Retrieval and Ranking
  - Efficient Candidate Retrieval
  - Deep Ranking Models
- **Language Generation for Search Assistance**
  - **Auto Completion**
  - **Query Reformulation**
  - **Spell Correction**

Language Generation for Search Assistance

- Auto Completion
- Query Reformulation
- Spell Correction

Common

- Goal: improve user experience by interacting with users
- NLP: Language generation

Difference

- Character/word modeling
- Generation models:
  - language modeling, seq2seq

Language Generation for Search Assistance

- Auto Completion
  - partial seq to seq
- Query Reformulation
  - word-level seq to seq
- Spell Correction
  - character-level seq to seq

Query Auto-Completion

- Problem statement: save user keystrokes by predicting the entire query

Challenges

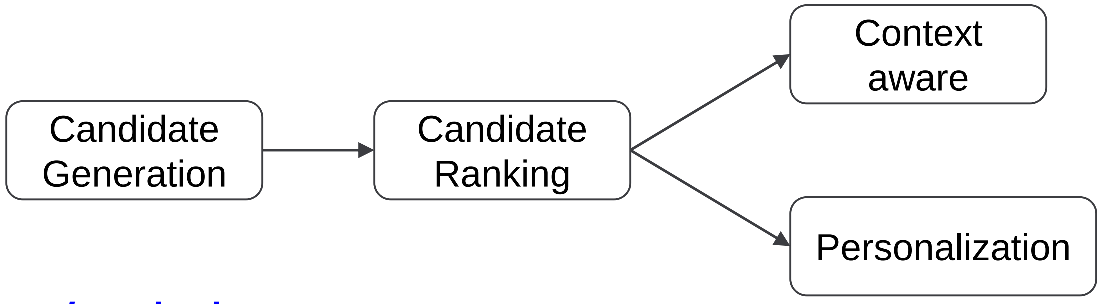
- Limited Context
  - Hard to extract features due to limited words in a query
- Latency
  - The search component with most strict requirement on latency

Candidate Generation and Ranking

- Traditional approach: 2-step approach

soft $\xrightarrow{\text{candidate generation}}$ *softbank*, *softball*, *softmax*, *software engineer*, …… $\xrightarrow{\text{candidate ranking}}$ *software engineer salary*, *soft skills*, *softball*, *softbank*, ……

Agenda



Traditional methods

Candidate Generation

- Collect completed queries and associated frequency from search log
- Efficiently retrieve most frequent queries starting with the prefix
  - Using a trie data structure

soft $\xrightarrow{\text{candidate generation}}$

softbank	100
softball	50
softmax	40
software engineer	35
…...	

Candidate Generation for Rare Prefix

(Mitra & Craswell, 2015)

- No such prefix in search log
  - “*cheapest flights from seattle to*”

“*cheapest flights from seattle to*” → “*to*” →

to dc	100
to sfo	50
to airport	40
to seattle	35
……	

Candidate Ranking

- Challenge: very few features can be extracted

soft $\xrightarrow{\text{candidate generation}}$

softbank
softball
softmax
software engineer
……

$\xrightarrow{\text{candidate ranking}}$

	frequency
software engineer salary	*100*
soft skills	*50*
softball	*20*
softbank	*15*
……	

Context-Aware

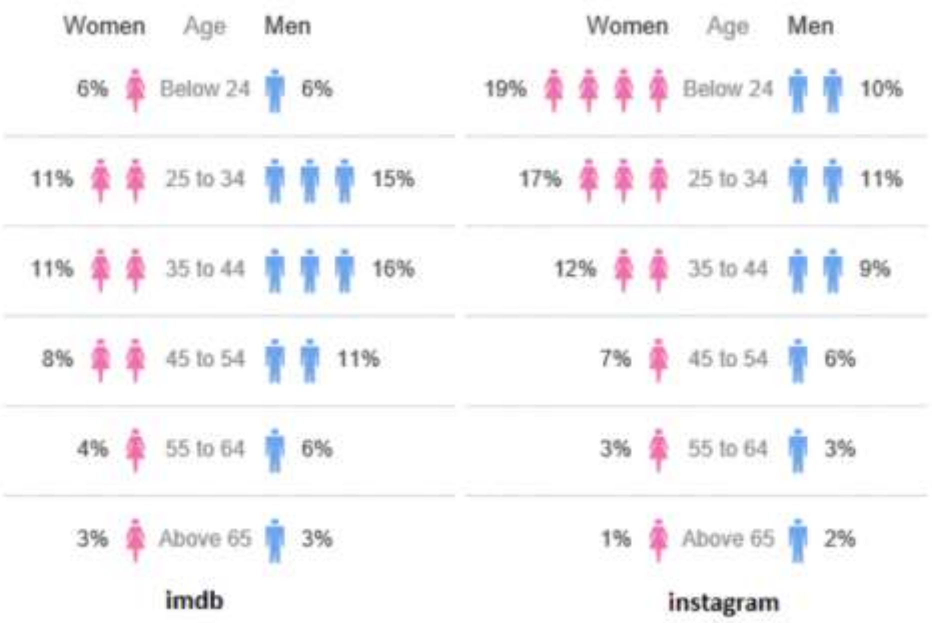
(Bar-Yossef & Kraus, 2011)

- Key idea: identify the candidates most similar to previous queries

	(candidates)			sim score
infant n →	*infant nike shoes*	→	*infant, nike, shoe, adidas, clothes...*	0.1
	infant nutrition		*infant, nutrition, baby, eat, food...*	0.8
	......		*......*	
	(previous queries)			
	baby eating disorder	→	*baby, eating, disorder, nutrition, food...*	
	......		*......*	

Personalization

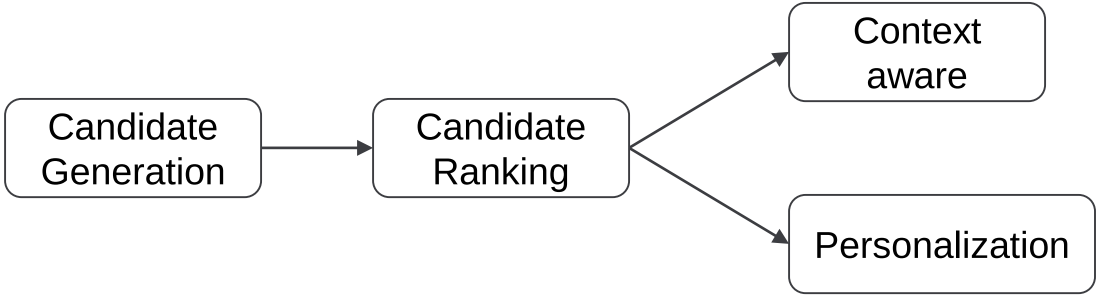
(Shokouhi, 2013)



query prefix is "i"

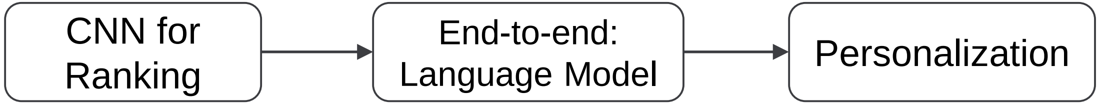
Feature list
Same age
Same gender
Same region
……

Agenda



Traditional methods

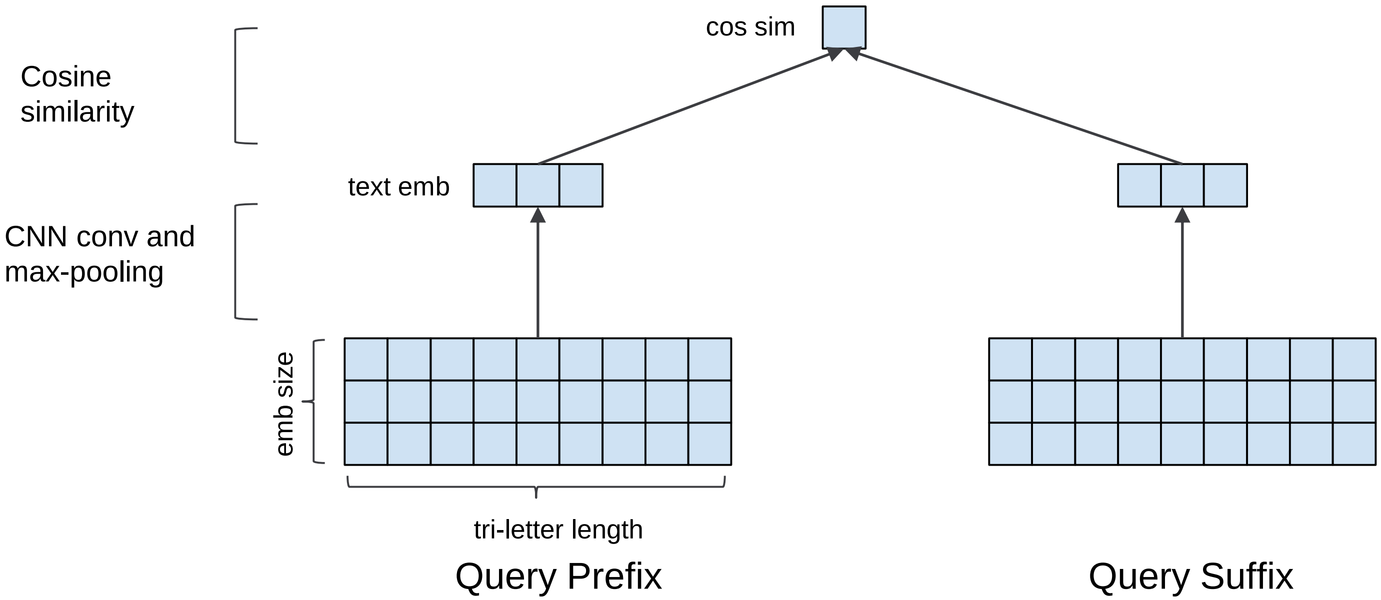
Deep NLP methods



Apply Deep Models in Ranking

(Mitra & Craswell, 2015)

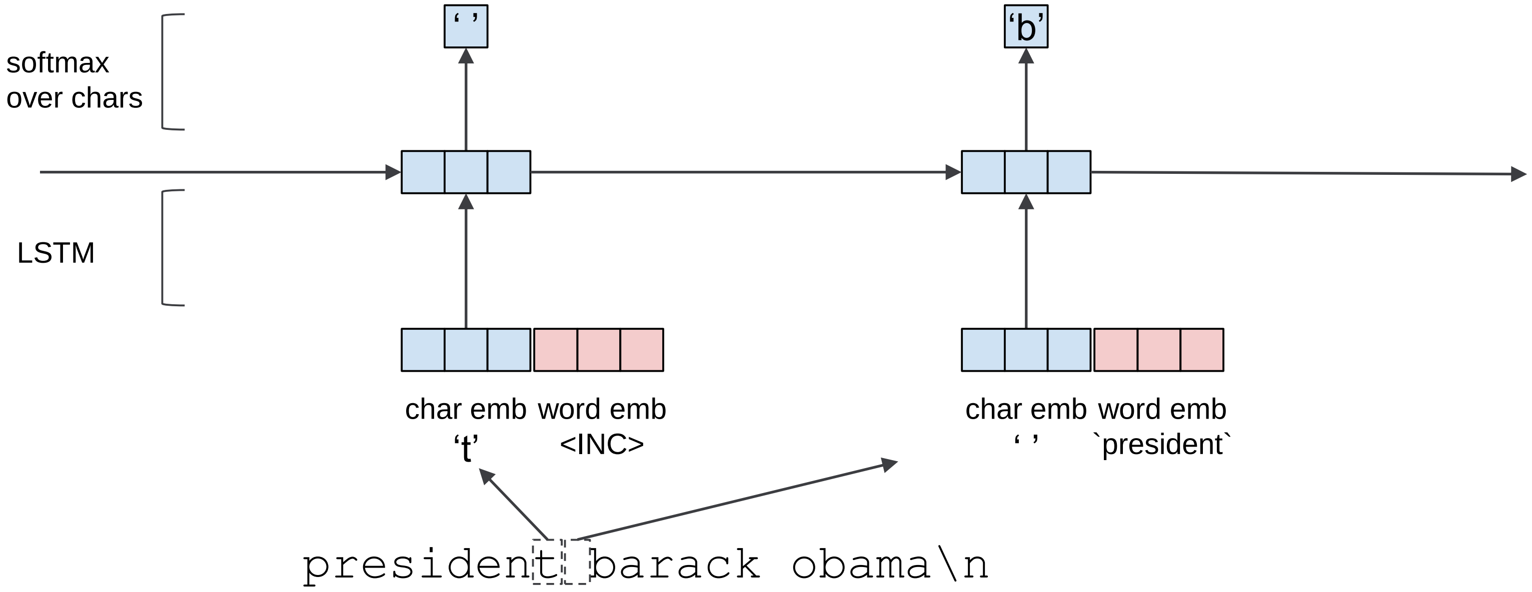
- Measuring the semantic coherence between prefix and suffix



Language Modeling for Auto-Completion

(Park & Chiba, 2017)

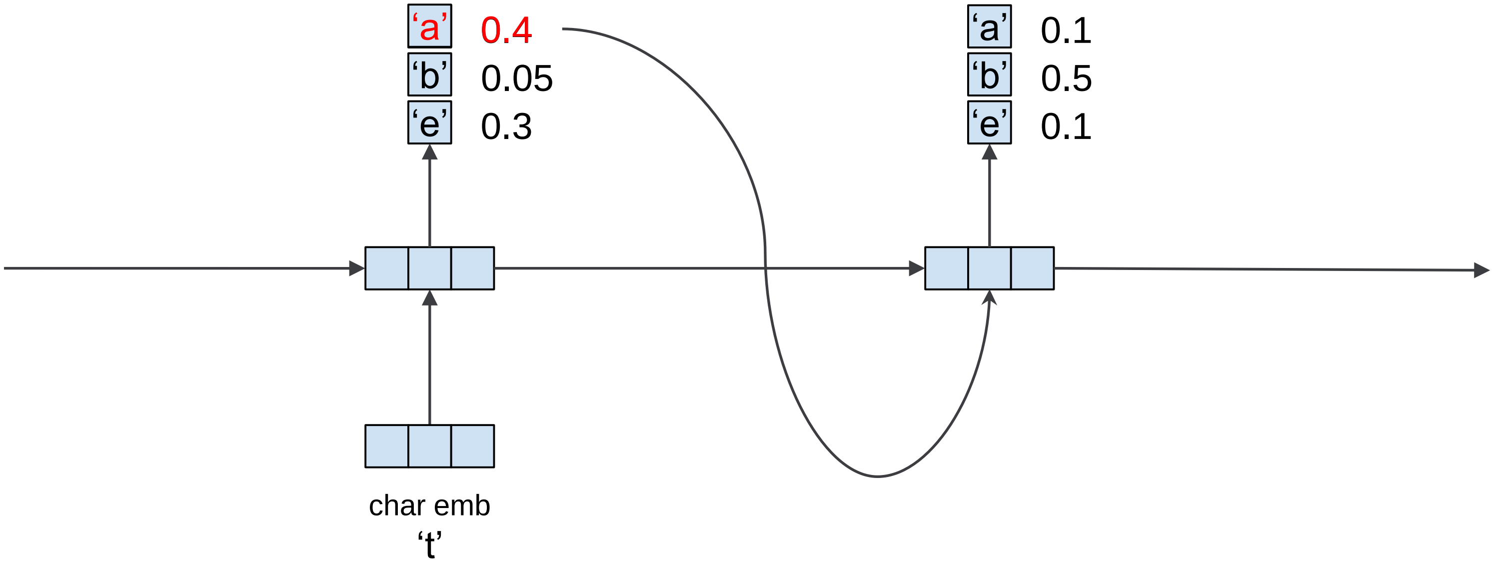
- Training: Character level language model + word embedding



Language Modeling for Auto-Completion

(Park & Chiba, 2017)

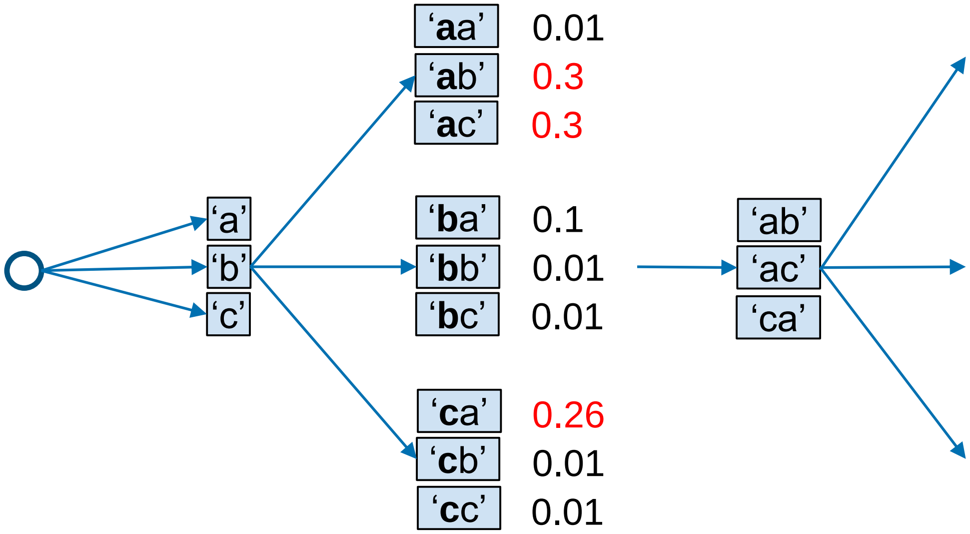
- Testing: Generating and ranking candidates at the same time



Language Modeling for Auto-Completion

(Park & Chiba, 2017)

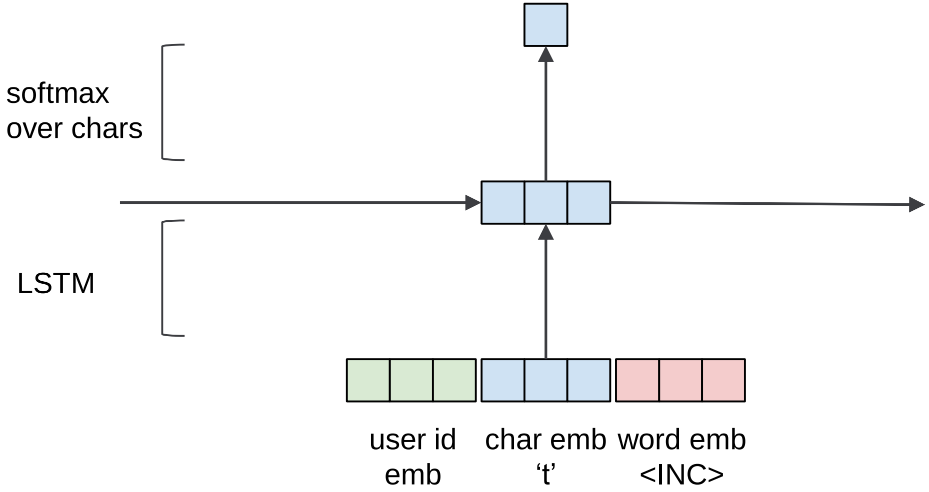
- Testing: Generating and ranking candidates at the same time
- ~~Greedy~~ vs beam search



Personalization

(Fiorin & Lu, 2018)

- Embeddings for User Ids



Query Auto-Completion: Summary

- Traditional methods: hard to extract features
- Deep language model framework:
  - Very flexible to incorporate personalized/contextualized information
  - An end-to-end solution
    - Train: all parameters are optimized together
    - Test: generation and ranking at the same time
  - Cons
    - Time-consuming
    - May generate wrong words

Reference

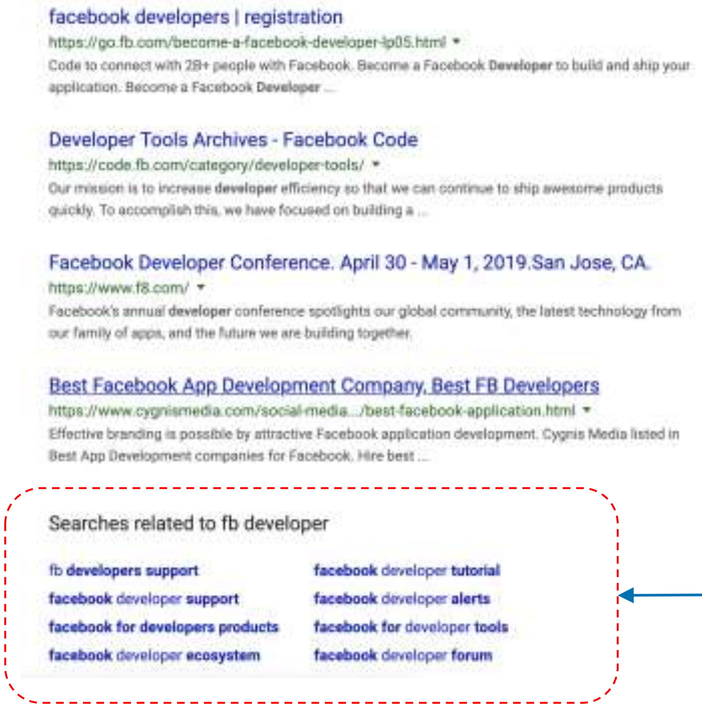
- Mitra, Bhaskar, and Nick Craswell. "Query auto-completion for rare prefixes." In Proceedings of the 24th ACM international on conference on information and knowledge management, pp. 1755-1758. ACM, 2015.
- Park, Dae Hoon, and Rikio Chiba. "A neural language model for query auto-completion." In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1189-1192. ACM, 2017.
- Bar-Yossef, Ziv, and Naama Kraus. "Context-sensitive query auto-completion." In Proceedings of the 20th international conference on World wide web, pp. 107-116. ACM, 2011.
- Shokouhi, Milad. "Learning to personalize query auto-completion." In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp. 103-112. ACM, 2013.
- Jaech, Aaron, and Mari Ostendorf. "Personalized language model for query auto-completion." arXiv preprint arXiv:1804.09661 (2018).
- Fiorini, Nicolas, and Zhiyong Lu. "Personalized neural language models for real-world query auto completion." arXiv preprint arXiv:1804.06439 (2018).

Language Generation for Search Assistance

- Auto Completion
  - partial seq to seq
- Query Reformulation
  - word-level seq to seq
- Spell Correction
  - character-level seq to seq

Query Reformulation

- Problem Statement: automatically reformulate the previous query



Search results for
"*fb developer*"

Reformulated queries for
"*fb developer*"

Agenda

collaborative filtering

Traditional methods

Traditional Approach: Collaborative Filtering

(Rida et al., 2012)

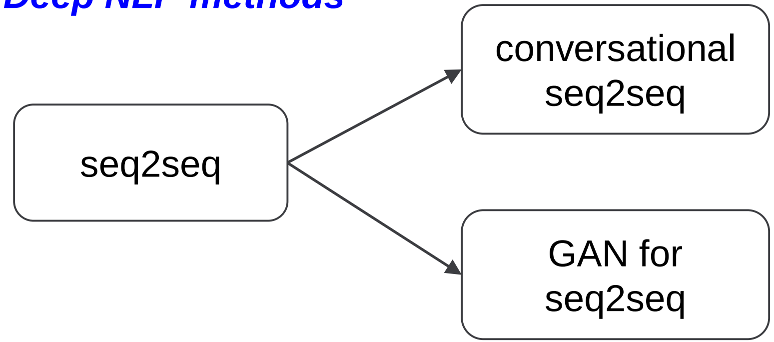
- Collect query pairs issued by the same user from search log
- Treat each query as an ID, and build a query-to-query matrix
  - Value in the matrix is TF-IDF

Agenda

collaborative filtering

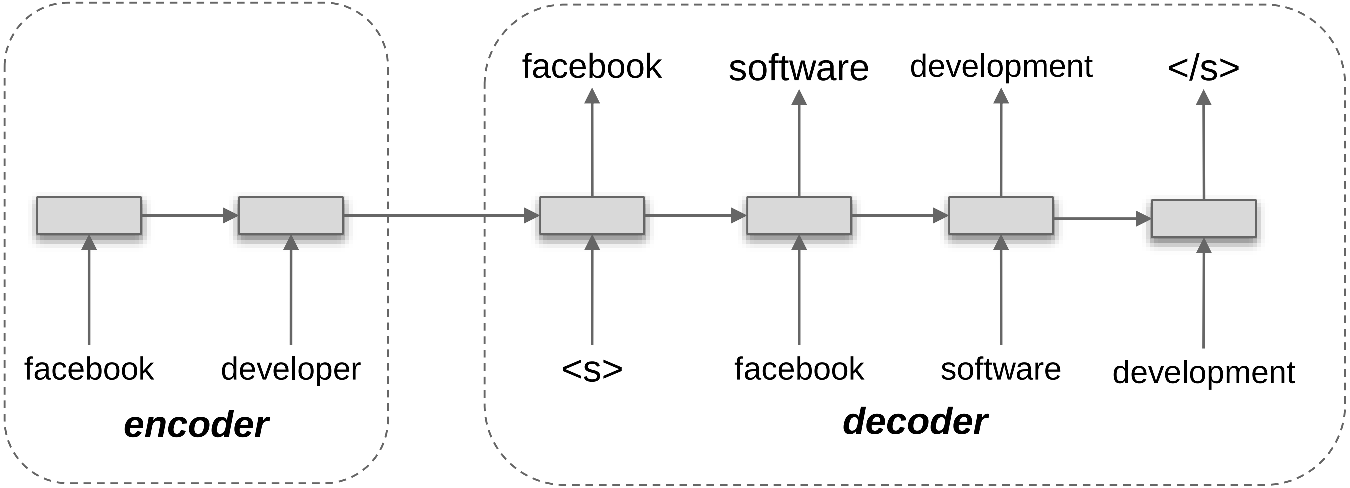
Traditional methods

Deep NLP methods



Query Reformulation as a Translation Task

- Sequence-to-sequence modeling (He et al, 2016)



- Directly modeling the words in a query

Conversational Query Reformulation

(Ren et al, 2018)

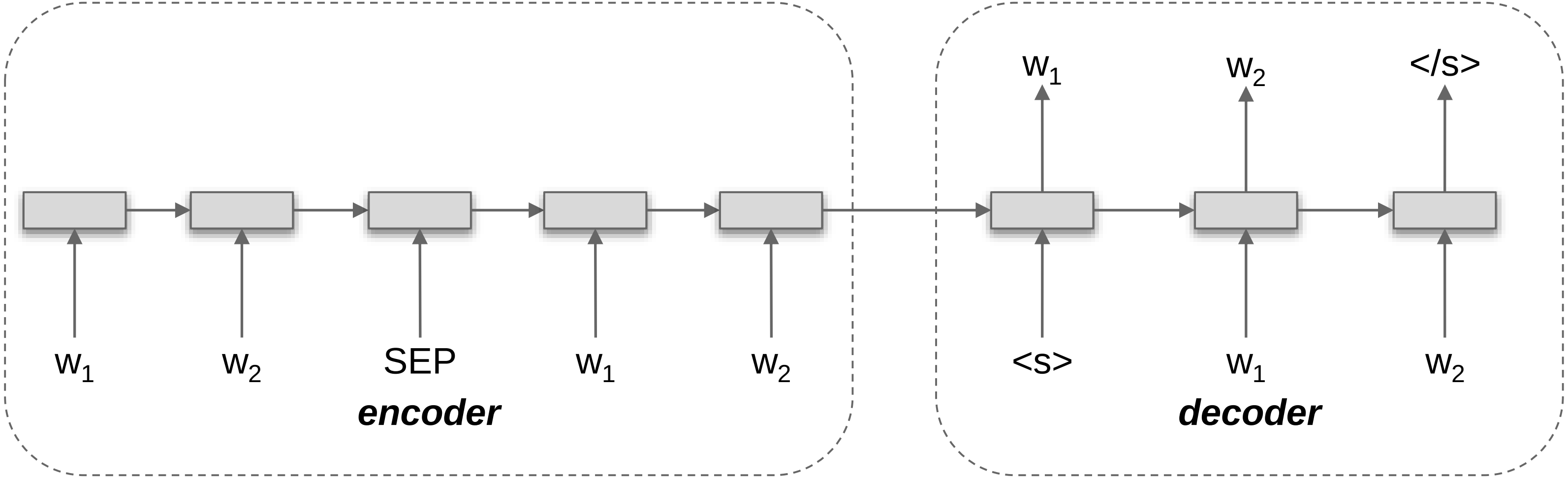
- Conversational queries
- Goal: summarize the conversation in one query

first query (q1)	**second query (q2)**	**summarized query (q3)**
when was California founded?	who is **its** governor?	who is California's governor?
California	population in 1990	population of California in 1990
how tall is kobe bryant?	**what about** Lebron James?	how tall is Lebron James?
when was the last summer Olympics?	and the winter one?	when was the last winter Olympics?

Conversational Query Reformulation

(Ren et al, 2018)

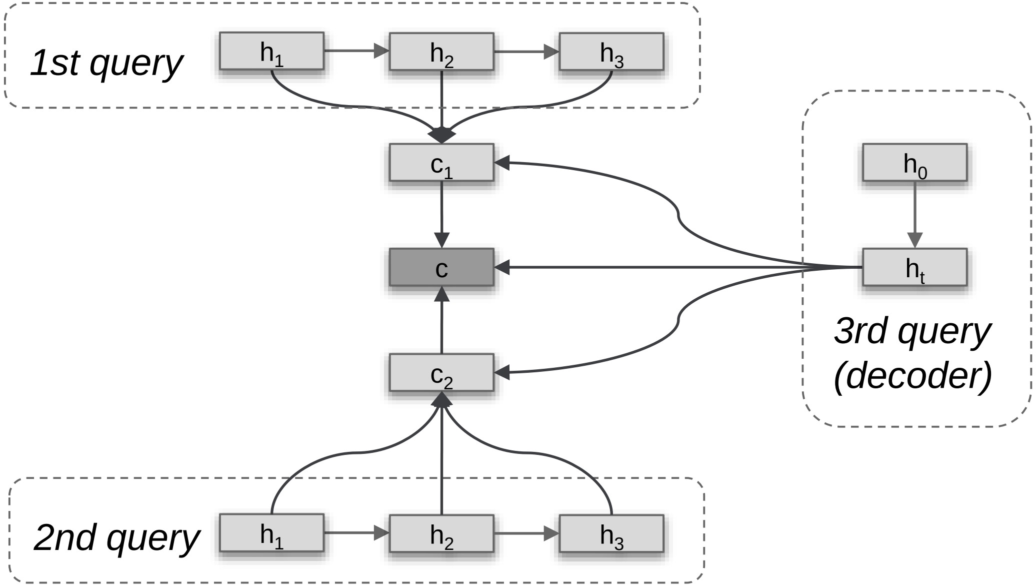
- 1st: concatenated seq2seq



Conversational Query Reformulation

(Ren et al, 2018)

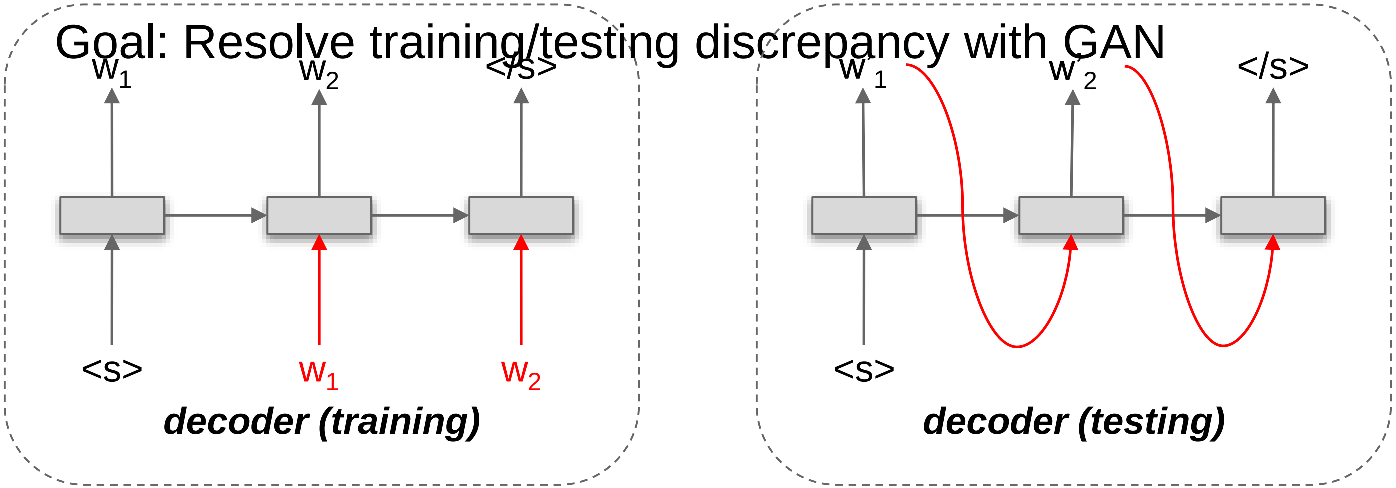
- 2nd: Attention over attention



GAN for seq2seq

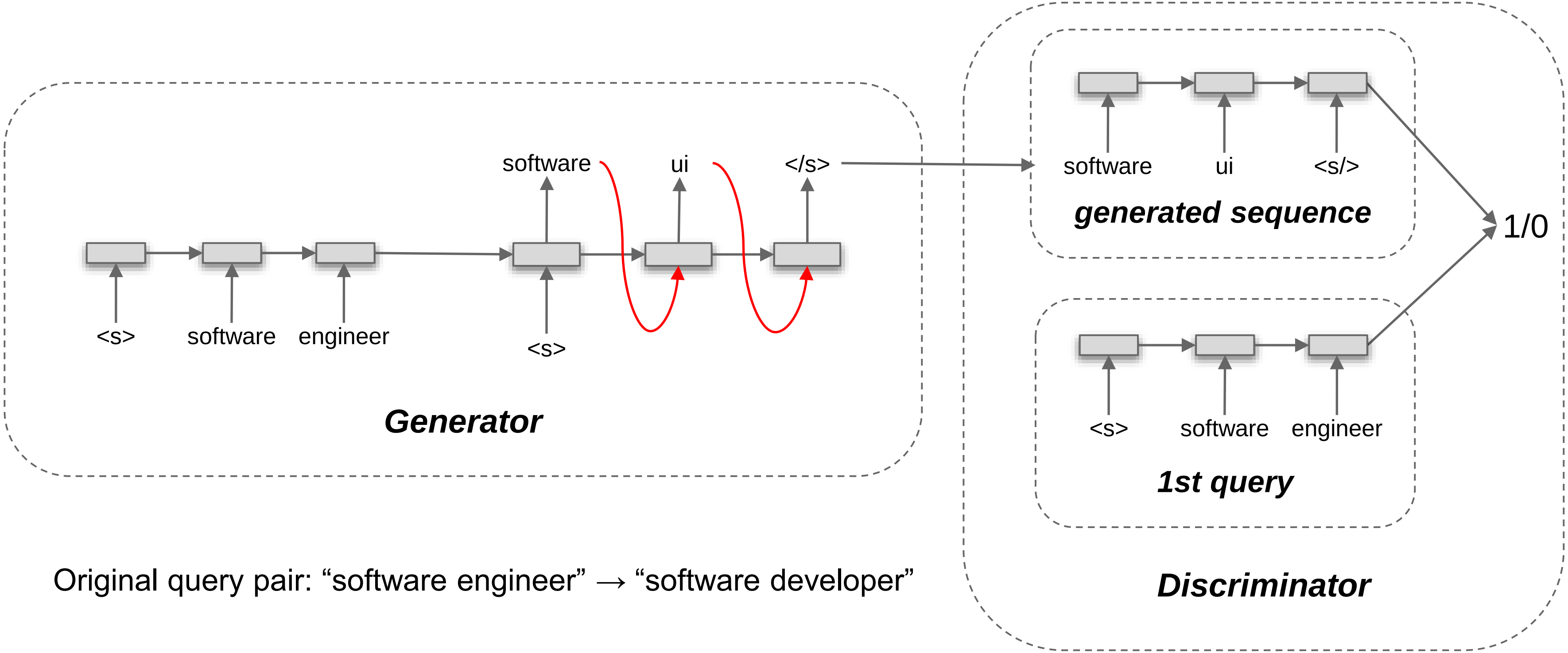
(Lee et al, 2018)

- Motivation: Discrepancy in decoder of seq2seq
  - Training: inputs are the gold-standard words
  - Testing: inputs are the previous predicted words
- Goal: Resolve training/testing discrepancy with GAN



GAN for seq2seq

(Lee et al, 2018)



Original query pair: “software engineer” → “software developer”

Query Reformulation: Summary

- seq2seq framework:
  - Directly modeling the words
  - Very flexible to incorporate session information
  - Achieves great performance (no character modeling)

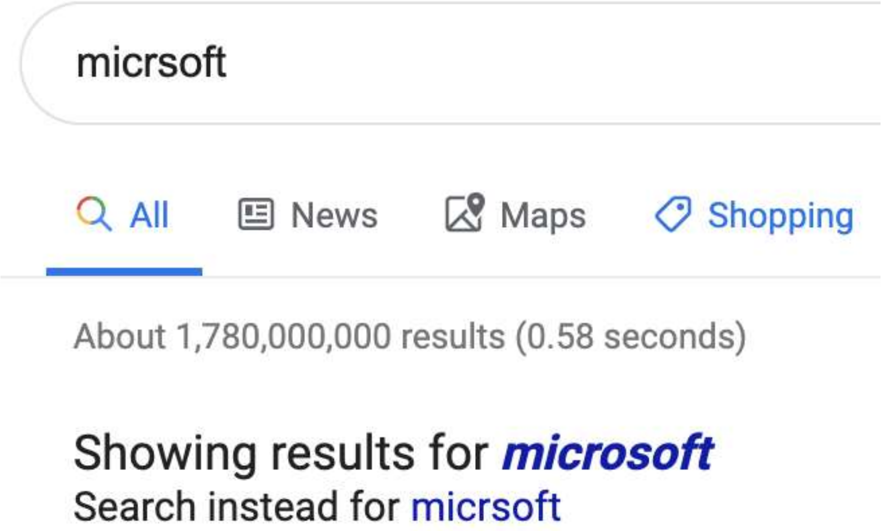
Reference

- Reda, Azarias, Yubin Park, Mitul Tiwari, Christian Posse, and Sam Shah. "Metaphor: a system for related search recommendations." In Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 664-673. ACM, 2012.
- He, Yunlong, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. "Learning to rewrite queries." In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 1443-1452. ACM, 2016.
- Ren, Gary, Xiaochuan Ni, Manish Malik, and Qifa Ke. "Conversational query understanding using sequence to sequence modeling." In Proceedings of the 2018 World Wide Web Conference, pp. 1715-1724. International World Wide Web Conferences Steering Committee, 2018.
- Lee, Mu-Chu, Bin Gao, and Ruofei Zhang. "Rare query expansion through generative adversarial networks in search advertising." In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 500-508. ACM, 2018.

Language Generation for Search Assistance

- Auto Completion
  - partial seq to seq
- Query Reformulation:
  - word-level seq to seq
- Spell Correction
  - character-level seq to seq

Spell Correction



Spell Correction

- Why spell correction:
  - Reduce the no results
- Challenge:
  - Many rare words (people/company names) look like spell errors
  - Modeling characters and words at the same time

query	**similar query**	**has error?**
tumblr	*tumble*	No
tumblw	*tumble*	Yes
galaxy s10e	*galaxy s10*	No
galaxy s10d	*galaxy s10*	Yes

Agenda

Candidate Generation for a word → Candidate generation for a query → Candidate ranking

Traditional methods

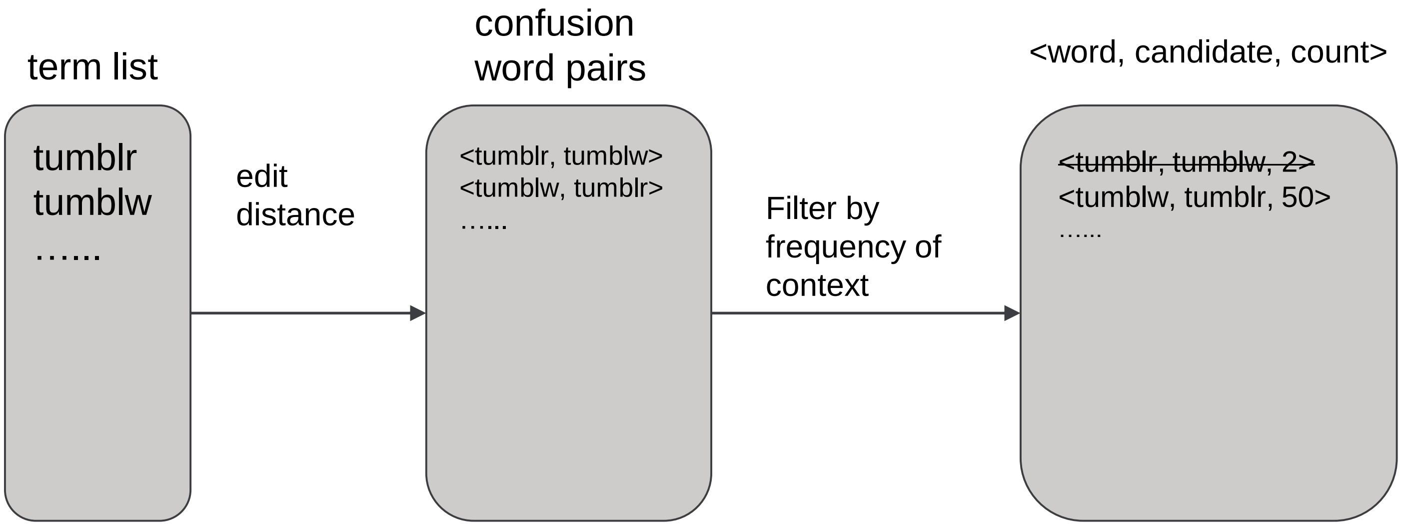
Candidate Generation for a Word

(Whitelaw et al 2009)

- Goal:
  - given “tumblw”, suggest “tumblr”, “tumble”...
  - given “tumblr”, suggest “tumble”, but not “tumblw”
- Key challenge: what is a legit word? (“tumblr” “tumblw”)
  - Coverage of a dictionary is not enough
- Solution: use statistics in web noisy data
  - Correct words appear more frequent than incorrect words

Candidate Generation for a Word

(Whitelaw et al 2009)



- For the context "social media X":
  - freq("social media tumblw") < freq("social media tumblr")

Candidate Generation for a Query

(Chen et al 2007)

earr pian kids

ear, era, ... *pain, pan, ...* *kids, kid, ...*

ear pain kids,
era pan kid,
ear pin kid,
…....

- Problem: query candidate size grows exponentially with # of words
- Solution: prune with language model
  - ***ear*** pian kids: 0.8
  - ~~***era*** pian kids: 0.1~~
  - earr ***pain*** kids: 0.9
  - earr ***pan*** kids: 0.7
  - ~~earr ***pien*** kids: 0.2~~
  - earr pian kid: 0.8

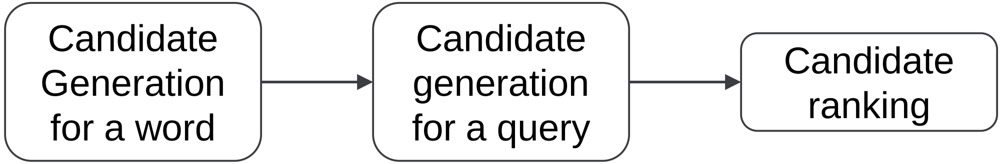
Candidate Ranking

(Li et al 2006, Chen et al 2007)

$$\underbrace{}_{}\operatorname{argmax}_c P(\overset{\text{candidate}}{c}|\overset{\text{query}}{q}) = \operatorname{argmax}_c P(q|c)P(c)$$

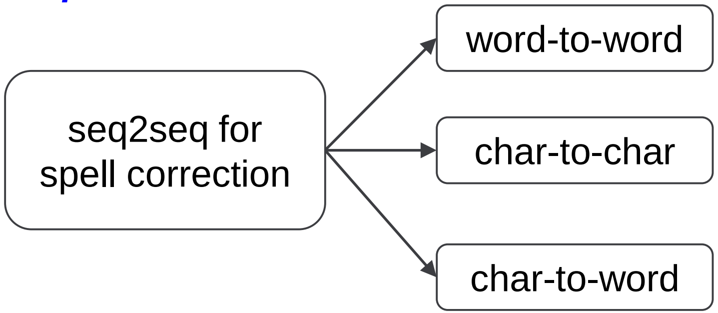
Feature Types	Examples
similarity $P(q\|c)$	Edit distance
	Frequency of user reformulation
	……
Likelihood $P(c)$	Language model score of the candidate
	Frequency of candidate terms appearing in the page titles
	……

Agenda



Traditional methods

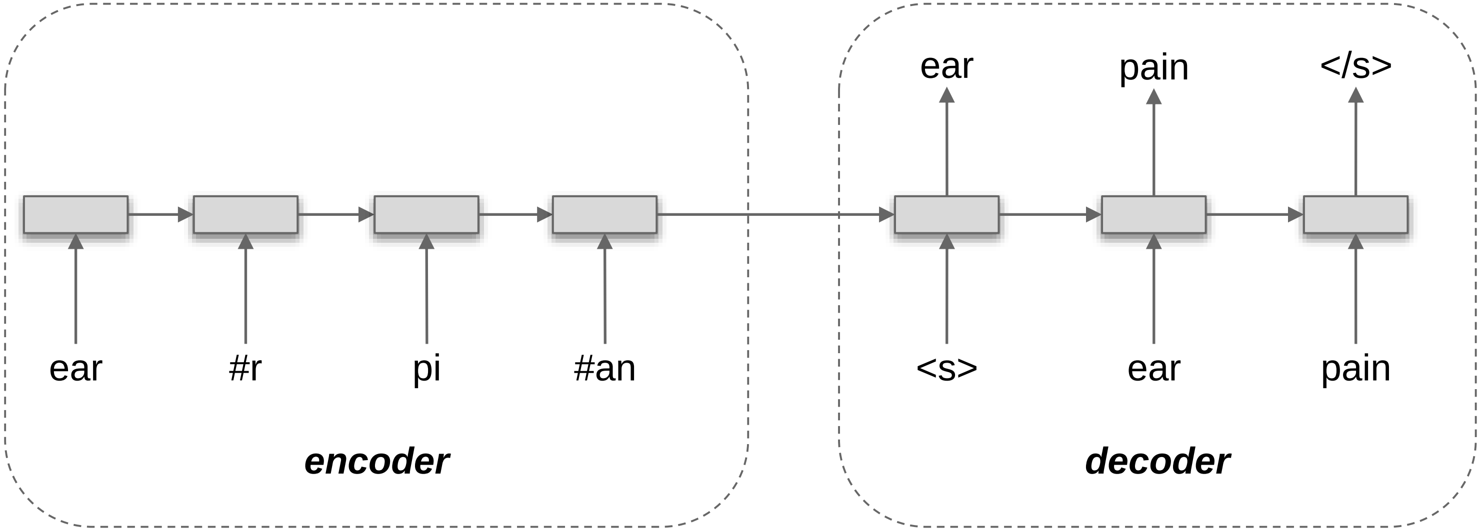
Deep NLP methods



Seq2seq for Spell Correction

(Ghosh and Kristensson, 2017, Zhou et al., 2017)

- From subwords to subwords



Seq2seq for Spell Correction

(Ghosh and Kristensson, 2017, Zhou et al., 2017)

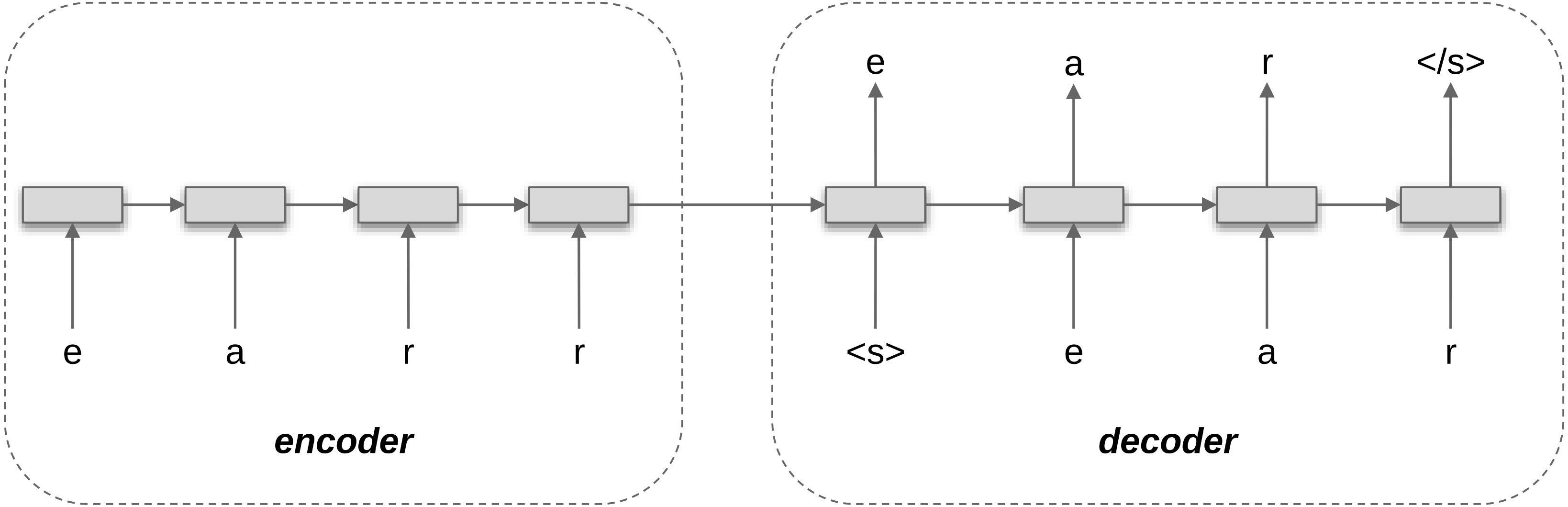
- From subwords to subwords
- Issue: subword is designed for texts without errors

	Normal texts	**Spell errors**
Example	*hunter → hunt #er*	*hunetr → hu #net #r*
Subword semantics	*relevant*	*irrelevant*

Seq2seq for Spell Correction

(Ghosh and Kristensson, 2017, Zhou et al., 2017)

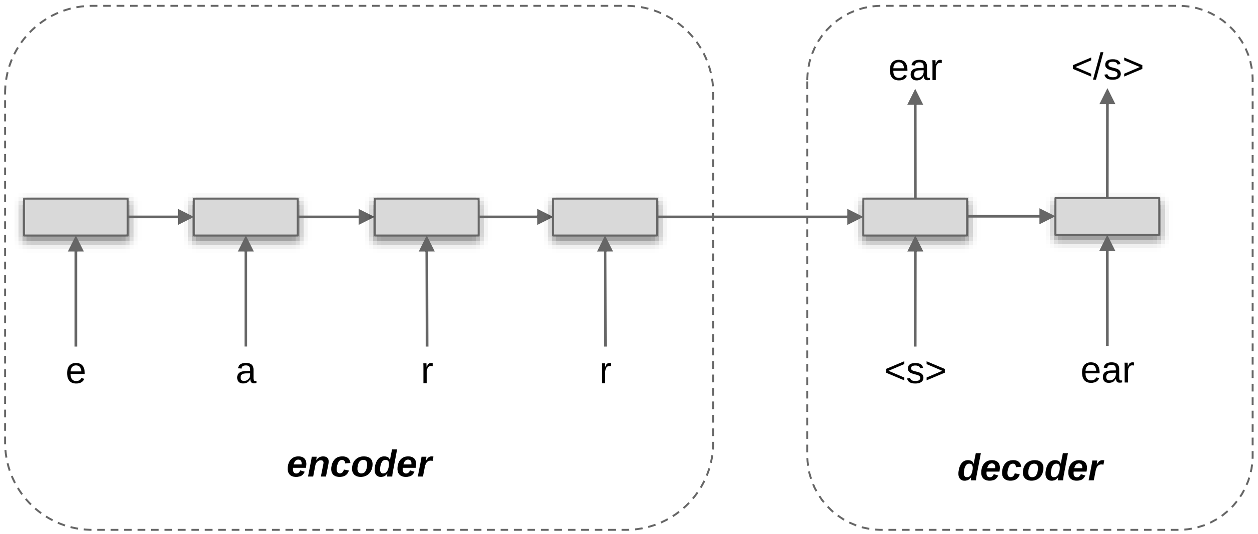
- From characters to characters
- Issue: on the decoder, no word information
  - Might produce words with wrong spelling



Seq2seq for Spell Correction

(Ghosh and Kristensson, 2017, Zhou et al., 2017)

- From characters to words
  - Most popular structure
  - Can leverage pretrained language model



Reference

- Hasan, Saša, Carmen Heger, and Saab Mansour. "Spelling correction of user search queries through statistical machine translation." In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 451-460. 2015.
- Yingbo Zhou, Utkarsh Porwal, Roberto Konow. "Spelling Correction as a Foreign Language." arXiv. 2017.
- Ghosh, Shaona, and Per Ola Kristensson. "Neural networks for text correction and completion in keyboard decoding." arXiv preprint arXiv:1709.06429 (2017).
- Chen, Qing, Mu Li, and Ming Zhou. "Improving query spelling correction using web search results." In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 181-189. 2007.
- Li, Mu, Yang Zhang, Muhua Zhu, and Ming Zhou. "Exploring distributional similarity based models for query spelling correction." In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 1025-1032. Association for Computational Linguistics, 2006.
- Whitelaw, Casey, Ben Hutchinson, Grace Y. Chung, and Gerard Ellis. "Using the web for language independent spellchecking and autocorrection." In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2, pp. 890-899. Association for Computational Linguistics, 2009.

Language Generation for Search Assistance: Summary

	Traditional methods	**deep NLP methods**
Candidate generation	Rule based	End-to-end solution Language modeling
Candidate ranking	Few features	
Latency	Low	High

Agenda

1 Introduction

2 Deep Learning for Natural Language Processing

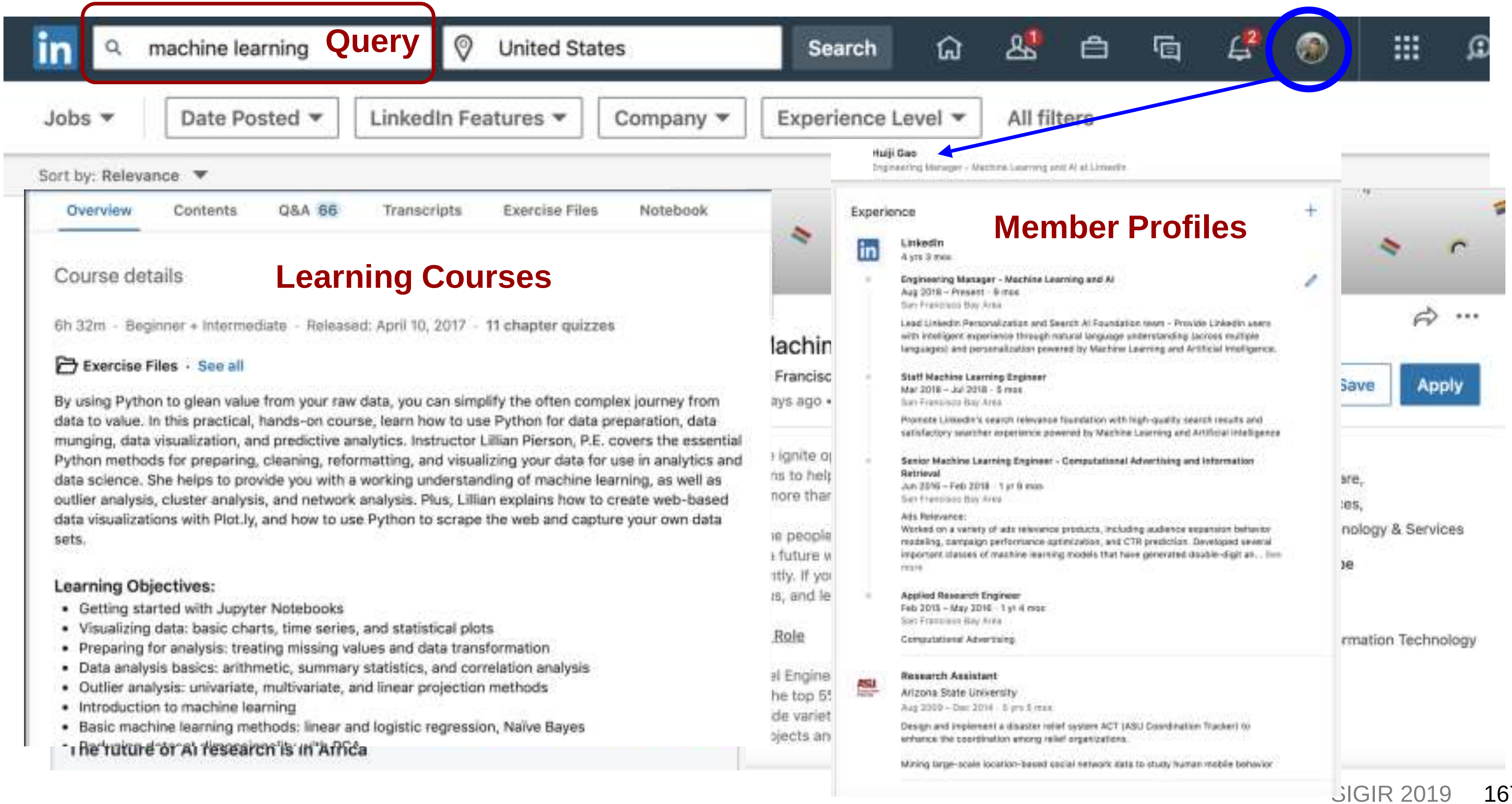
3 Deep NLP in Search Systems

4 **Real World Examples**

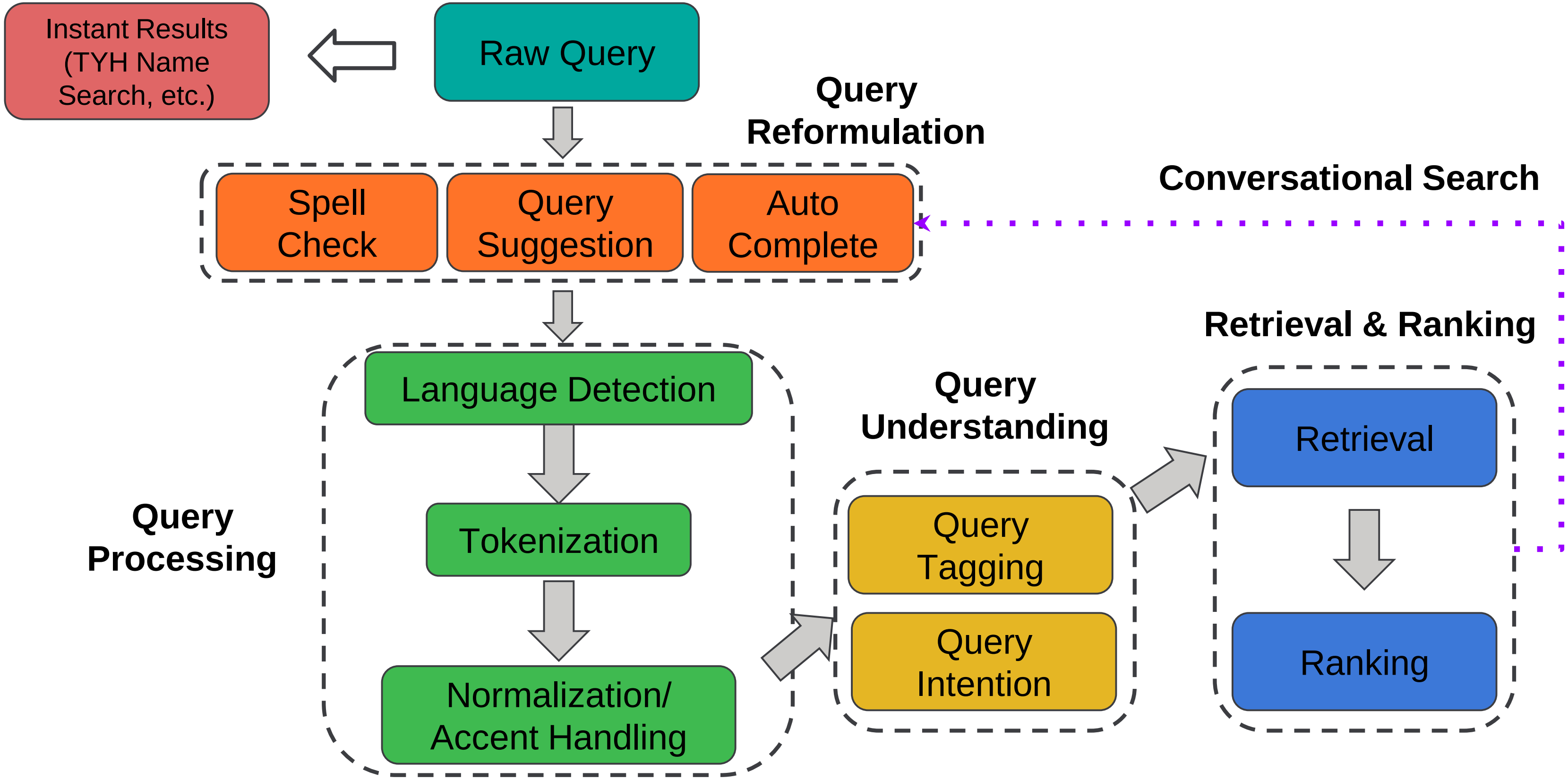
Deep NLP in Search Systems - Real World Examples

Huiji Gao

Natural Language Data in LinkedIn Search Systems



LinkedIn Search Ecosystem



NLP in LinkedIn Search: Challenges

- **Data Ambiguity**
  - Short Query Text
    - "abc"

ABC News? ABC Stores?

  - No Strict Syntax
    - "bing search engineer"

"Bing Search, Engineer" **"Bing, Search Engineer"**

  - Strong Correlation to the Searcher
    - "looking for new jobs"

Job Seeker looks for jobs

Recruiter looks for candidates

- **Deep Semantics**
  - Representations for query & document w.r.t. search intent, entities, topics
    - "Engineering Openings" -> Job Posts

Deep NLP in LinkedIn Search: Challenges

- **Complicated Search Ecosystem**
  - Query suggestion affects both recall and precision in downstream retrieval and ranking.
  - Query tagging needs to be compatible with indexing and align with ranking features.

- **Product Oriented Model Design**
  - Design deep NLP algorithms for specific search components
  - Consider business rules, post filters, results blender, user experience, etc

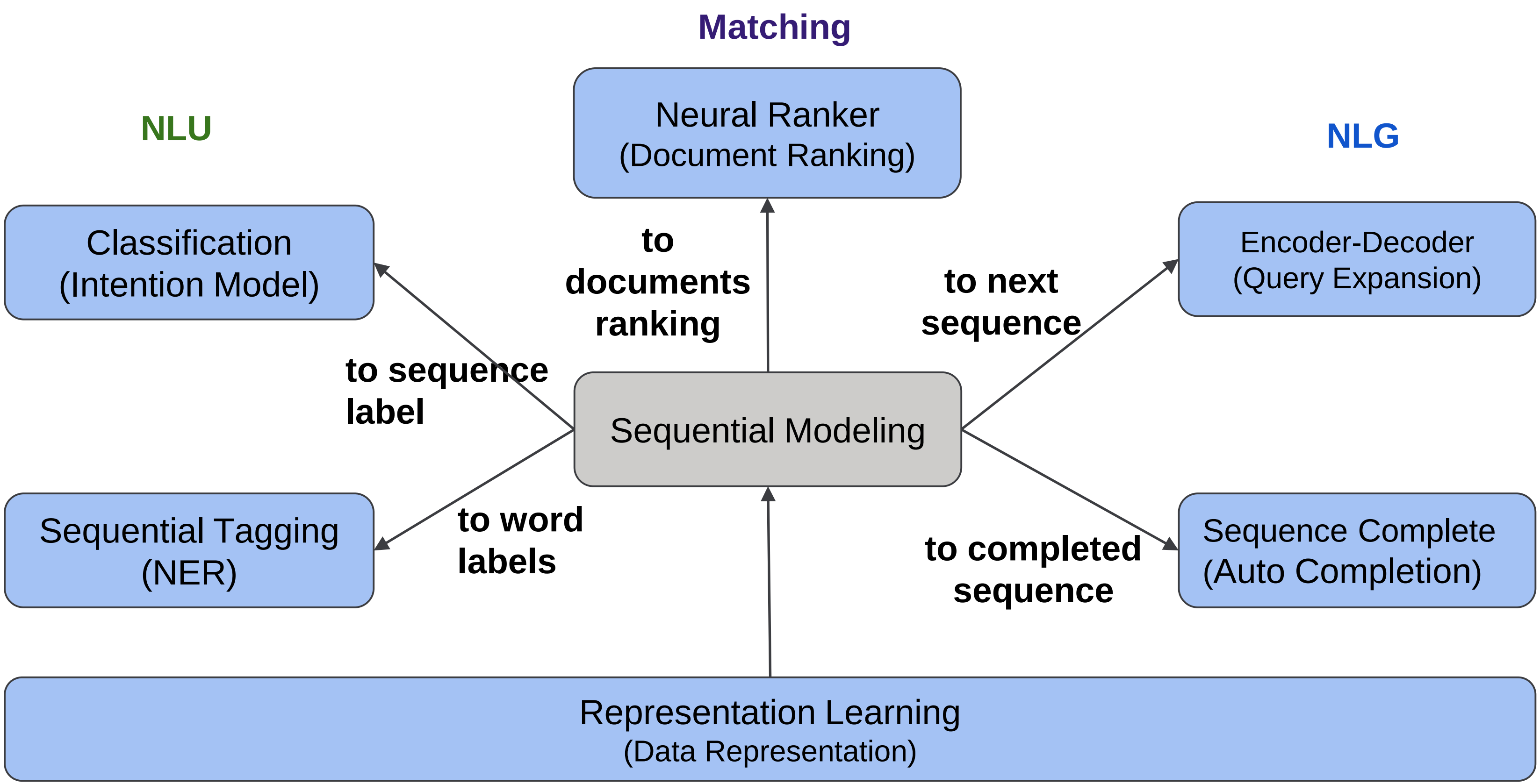
- **Online Latency**
  - Serving deep NLP models with product latency restriction

Applying Deep NLP in LinkedIn Search

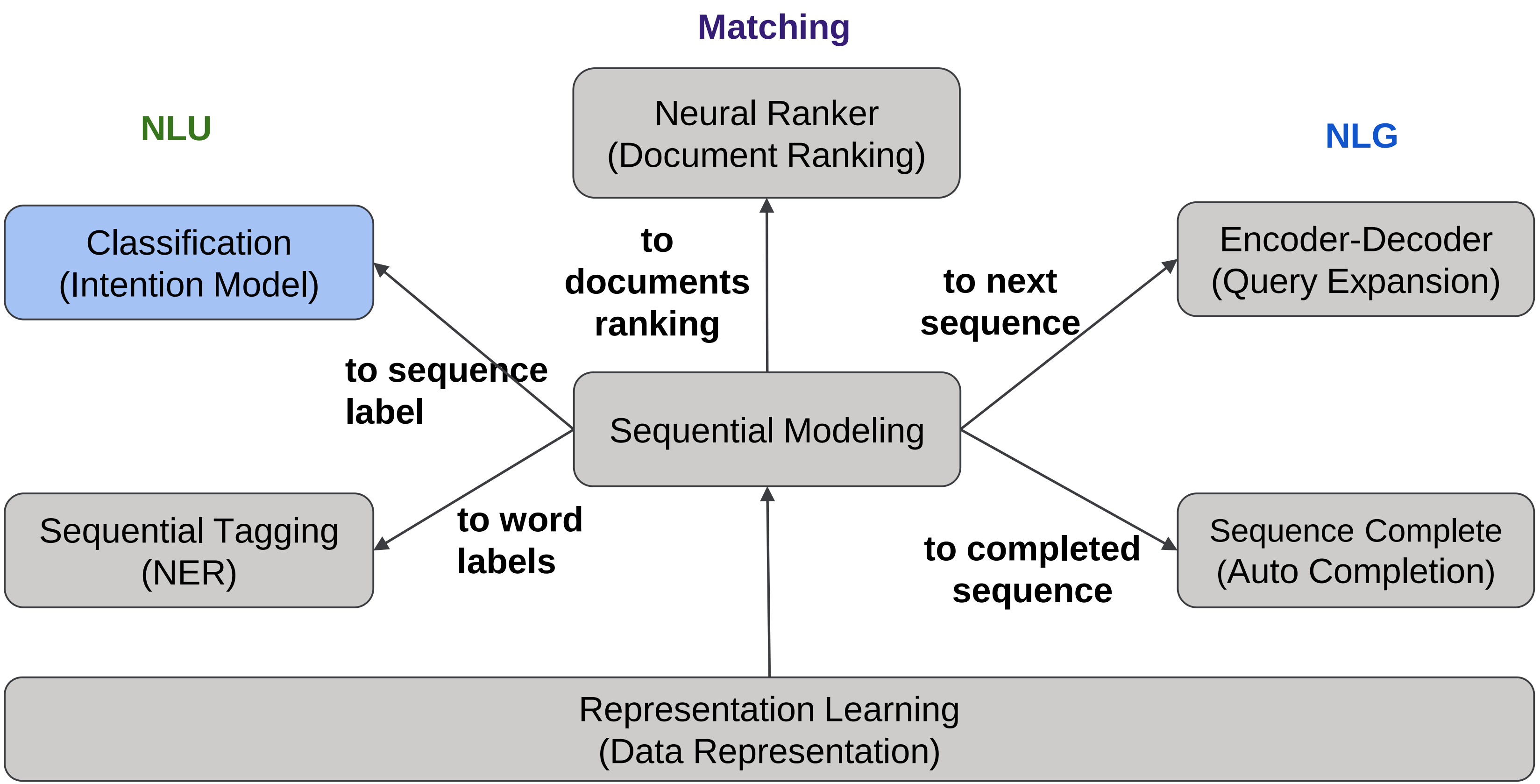
- **Feature Driven**
  - Representation Learning
    - Using features generated from deep learning models
      e.g., word embedding

- **Model Driven**
  - Power product features directly with deep learning models
    - CNN/LSTM/Seq2seq/GAN/BERT based deep NLP models

Deep Learning for Natural Language Processing



Deep Learning for Natural Language Processing



Query Intention Model: Goal

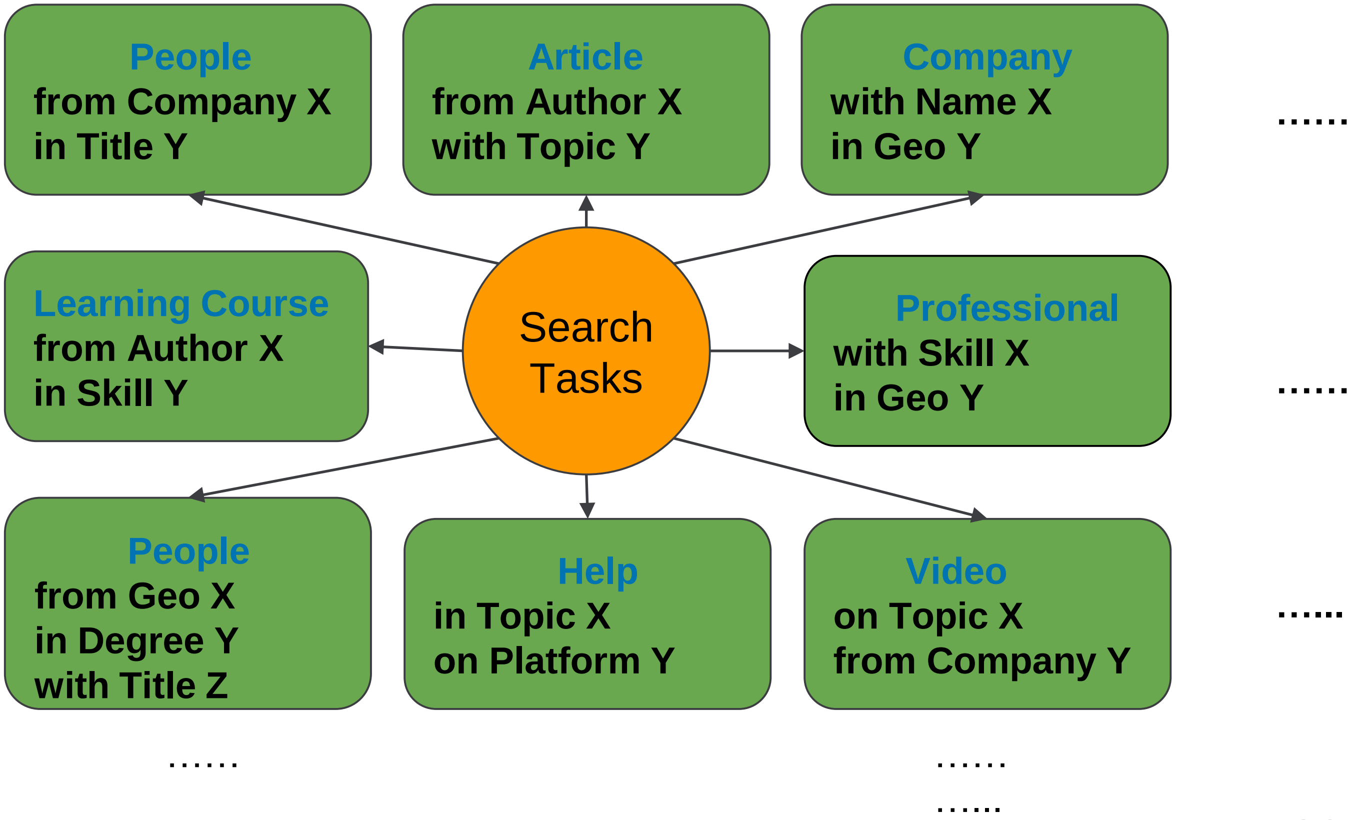
Query: LinkedIn Software Engineer

- **Output of Query Intention Model**
  - Search Vertical Prediction
    - **People, Job Posts, Articles ...**
  - Properties Associated with the Vertical
    - **Extracted** from raw query **OR** **Inferred** from other information

The searcher is looking for:

0.99	0.06	0.03	
People from "**LinkedIn**" as a "**Software Engineer**"	**Job Post** from "**LinkedIn**" on "**Software Engineer**" position	**Article** from "**LinkedIn**" on "**Software Engineer**" topic	……

Query Intention Model: Task Oriented Intention Prediction



Query Intention: Member Search Footprint

Query Intention Model: Goal

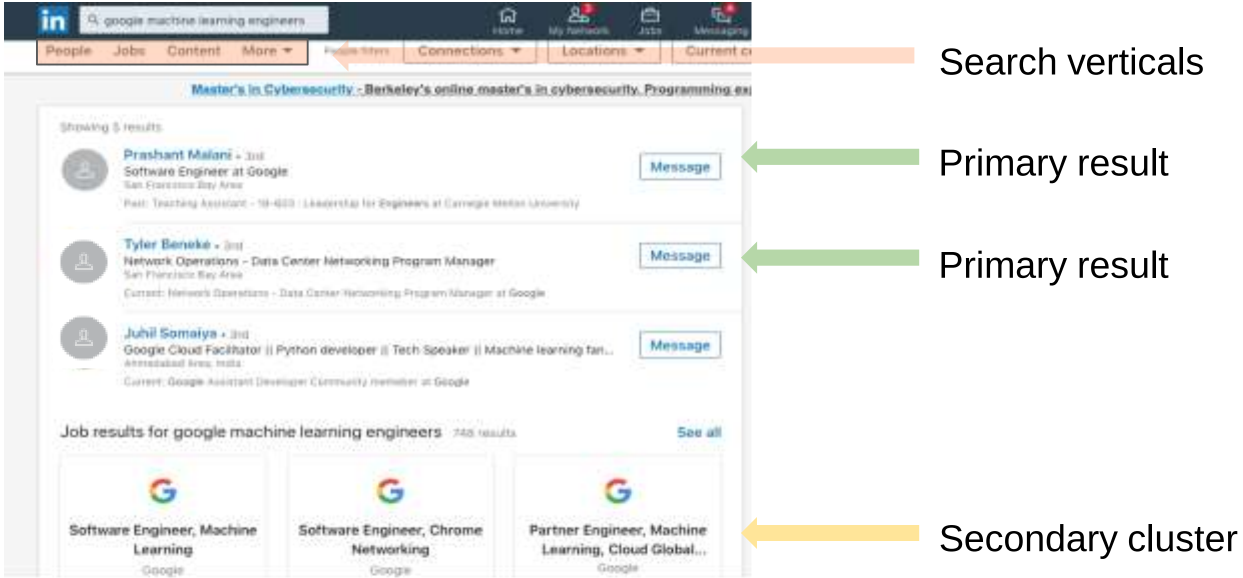
Query: LinkedIn Software Engineer

- **Output of Query Intention Model**
  - Vertical Prediction
    - **People, Job Posts, Articles ...**
  - Properties Associated with the Vertical
    - **Extracted** from raw query **OR Inferred** from other information

The searcher is looking for:

0.99	0.06	0.03	
People from "**LinkedIn**" as a "**Software Engineer**"	**Job Post** from "**LinkedIn**" on "**Software Engineer**" position	**Article** from "**LinkedIn**" on "**Software Engineer**" topic	……

Query Intention on Search Blending

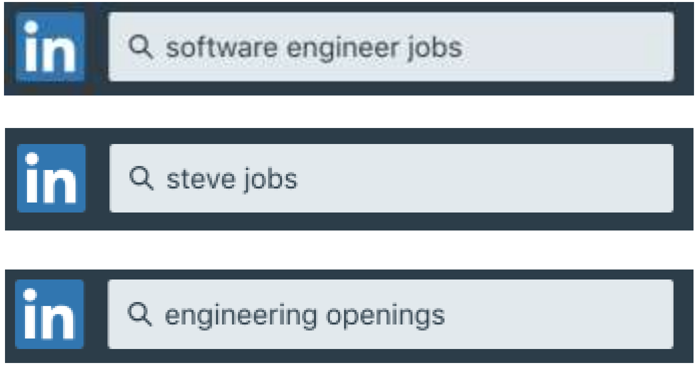


Goal:

- To understand the vertical preference of user on LinkedIn search

Query Intention Model: Challenges

- Complicated Semantics



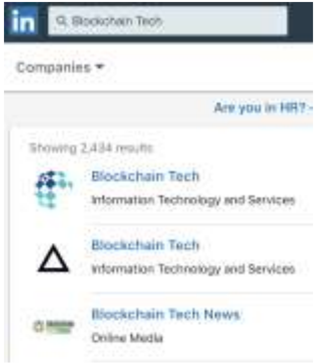
Job

Job?

Job!

Query Intention Model: Challenges

- Personalization
  - **Query:** Blockchain Tech
    - **Job seeker** looks for Blockchain Technology job
    - **Company** named Blockchain Tech
    - **Learning course** on Blockchain Tech
    - **Content** on Blockchain Technology
    - **Video** about blockchain technology
    - **Recruiter** looks for candidates with Blockchain tech skill
    - ……



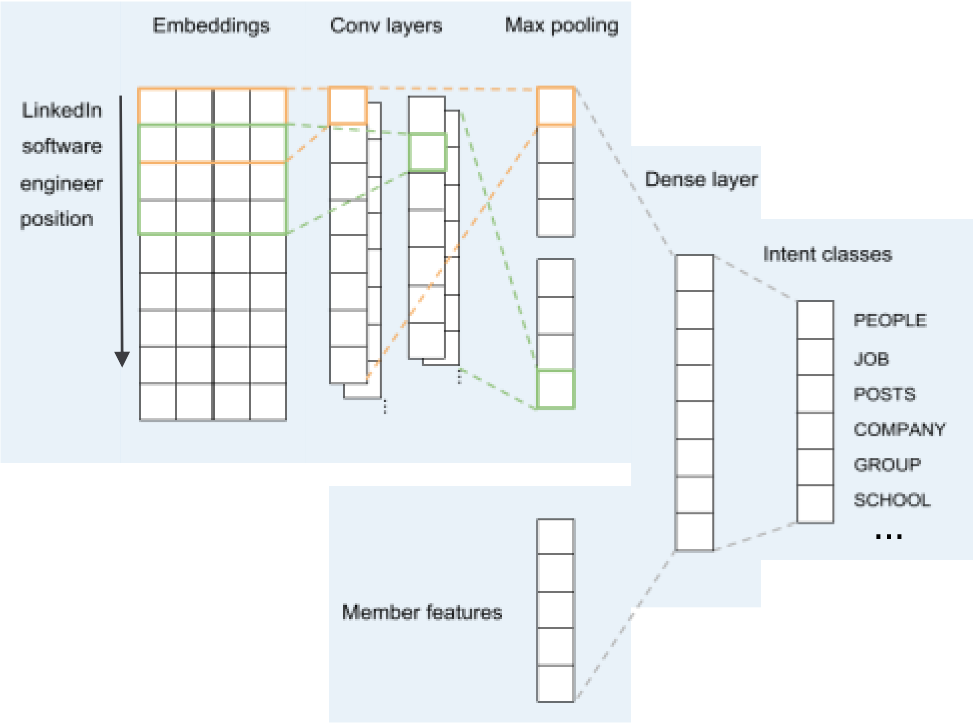
CNN Based Query Intention Model

CNN for Semantic Feature Extraction

- Word/query representations
- Generalization power
- Word n-gram patterns

Personalization

- Member-level Features



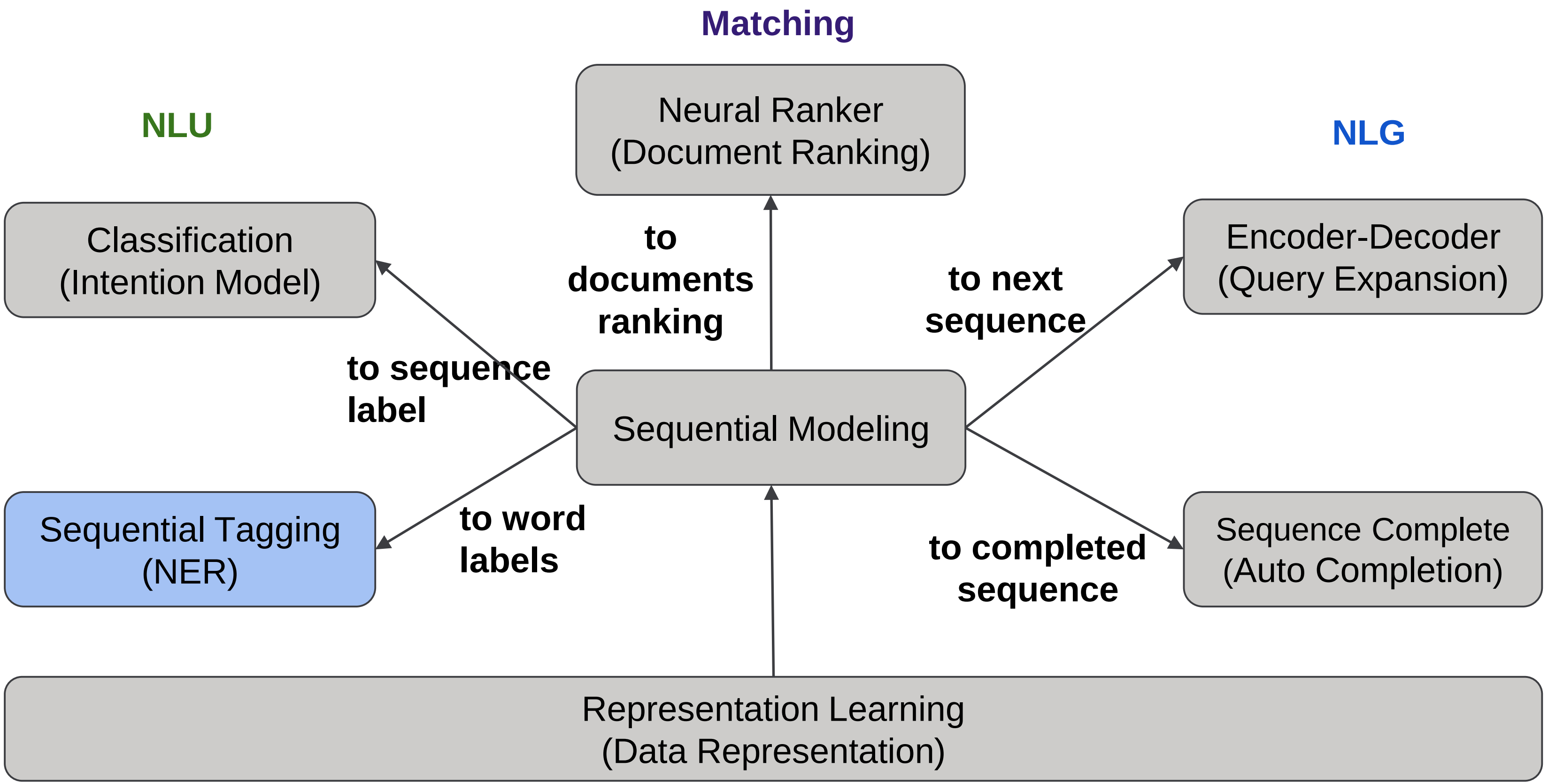
Query Intent - Experiment Results

- Offline Results

	Overall Accuracy	F1 on PEOPLE	F1 on JOB
Baseline (ML Model)	-	-	-
CNN	+2.9%	+11.9%	+1.7%

- Online Results
  - +0.65% JOB Ctr At 1 Serp
  - +0.90% Overall Cluster Ctr, +4.03% Cluster Ctr Via Entity Click

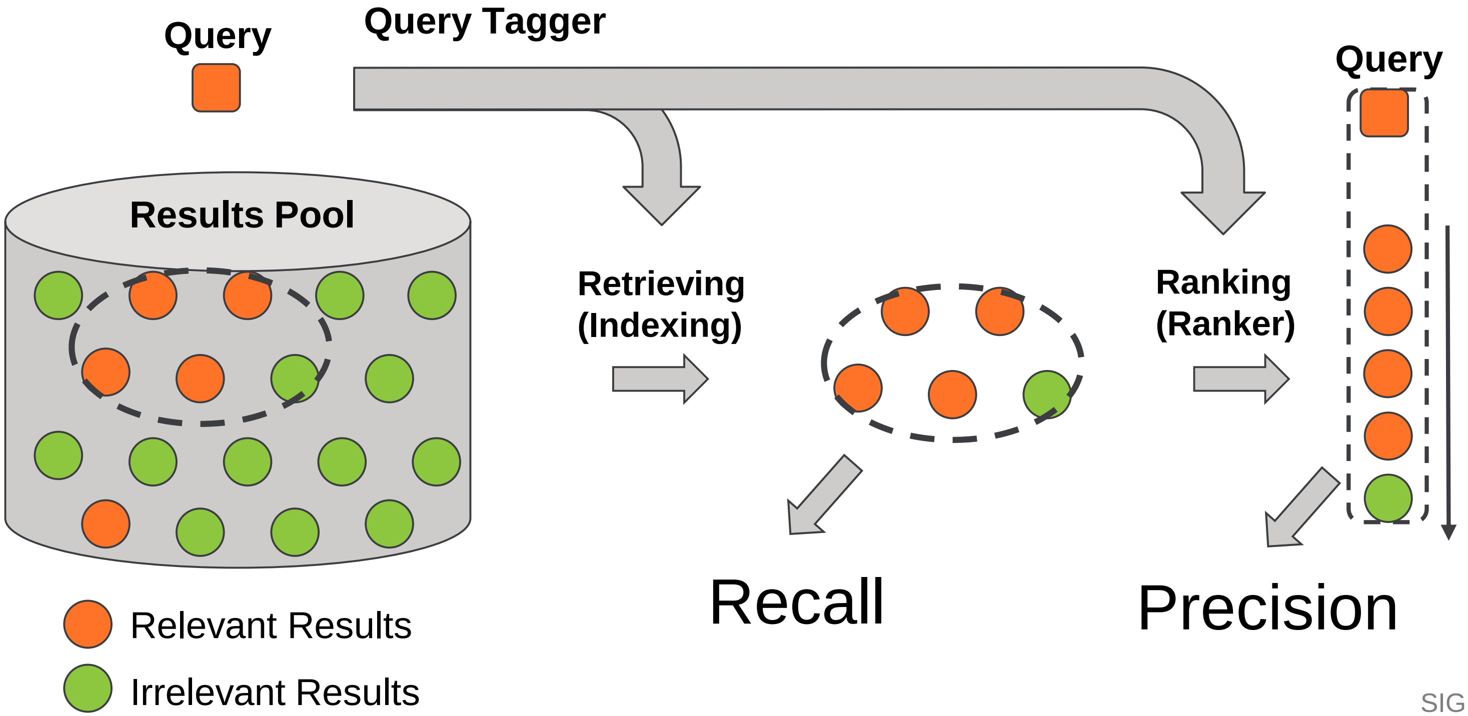
Deep Learning for Natural Language Processing



Query Tagger at LinkedIn

- **LinkedIn Search Engine**

Query: Mike LinkedIn Software Engineer



Search at LinkedIn

- **Understanding Queries with Query Tagger**

Query: Mike LinkedIn Software Engineer

  - **Query Tagger for Retrieval**

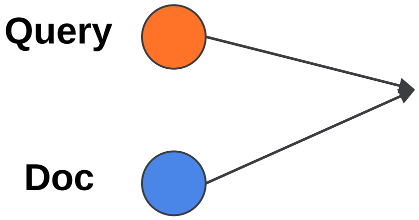
CN: company name
FN: first name
T: title

Mike LinkedIn Software Engineer
FN CN
T T

Index

FN:{mike}	Doc1, **Doc2**, **Doc4**, ...
CN:{linkedin}	**Doc2**, **Doc4**, Doc5, Doc6, ...
T:{Software}	**Doc2**, Doc3, **Doc4**, Doc7, ...
T:{Engineer}	Doc1, **Doc2**, **Doc4**, Doc9, ...

  - **Ranking Features**



IsFnMatch
IsLnMatch
IsTitleMatch
etc.

Natural Language Understanding: Query Tagger

LinkedIn	software	engineer	data	scientist	jobs
CN	T	T	T	T	O
B-CN	B-T	I-T	B-T	I-T	O

B-CN: beginning of a company name

I-CN: Inside of a company name

B-T: beginning of a job title

I-T: Inside of a job title

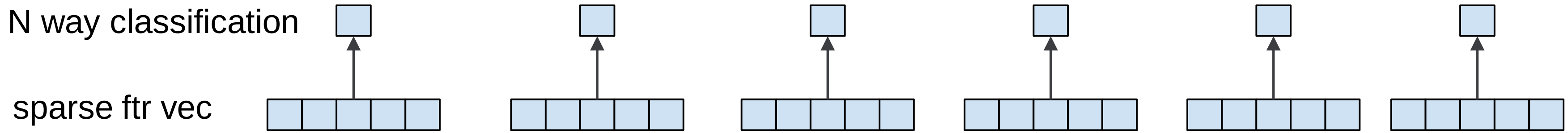
O: Not an entity

B-PN: beginning of person name

…

Query Tagger: Logistic Regression

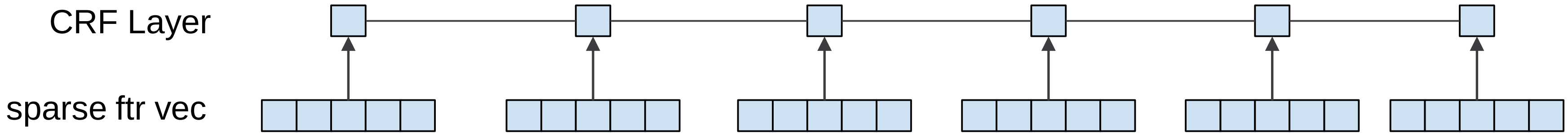
LinkedIn	software	engineer	data	scientist	jobs
B-CN	B-T	I-T	B-T	I-T	O



ftr 0: whether the current word is “linkedin”
ftr 1: whether the current word is “facebook”
…
ftr n: whether the next word is “software”
ftr n+1: whether the next word is “linkedin”
….

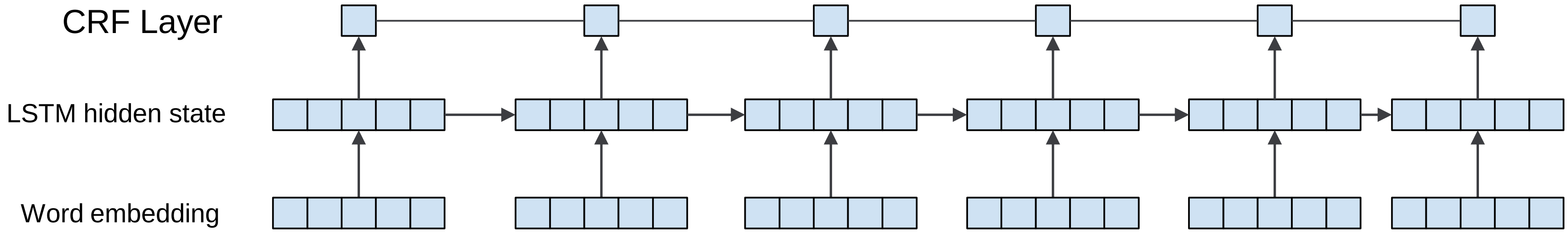
Query Tagger: CRF

LinkedIn	software	engineer	data	scientist	jobs
B-CN	B-T	I-T	B-T	I-T	O

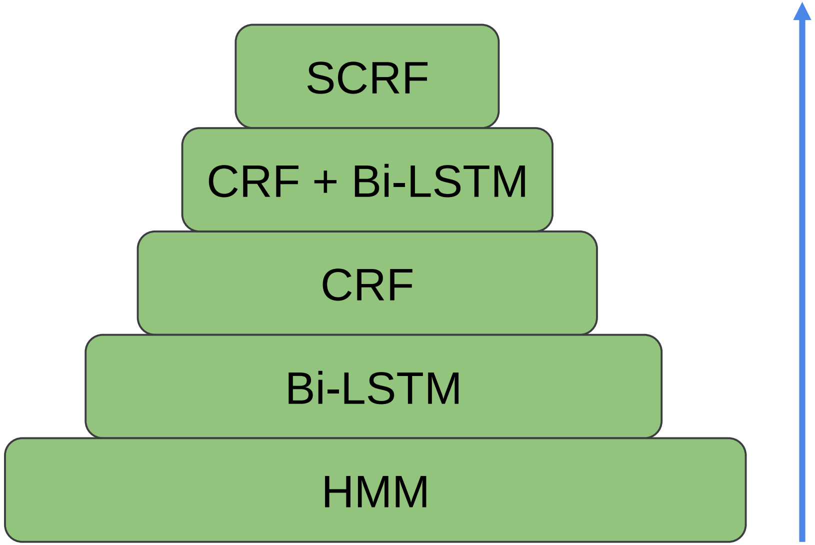


Query Tagger: CRF + LSTM

LinkedIn	software	engineer	data	scientist	jobs
B-CN	B-T	I-T	B-T	I-T	O

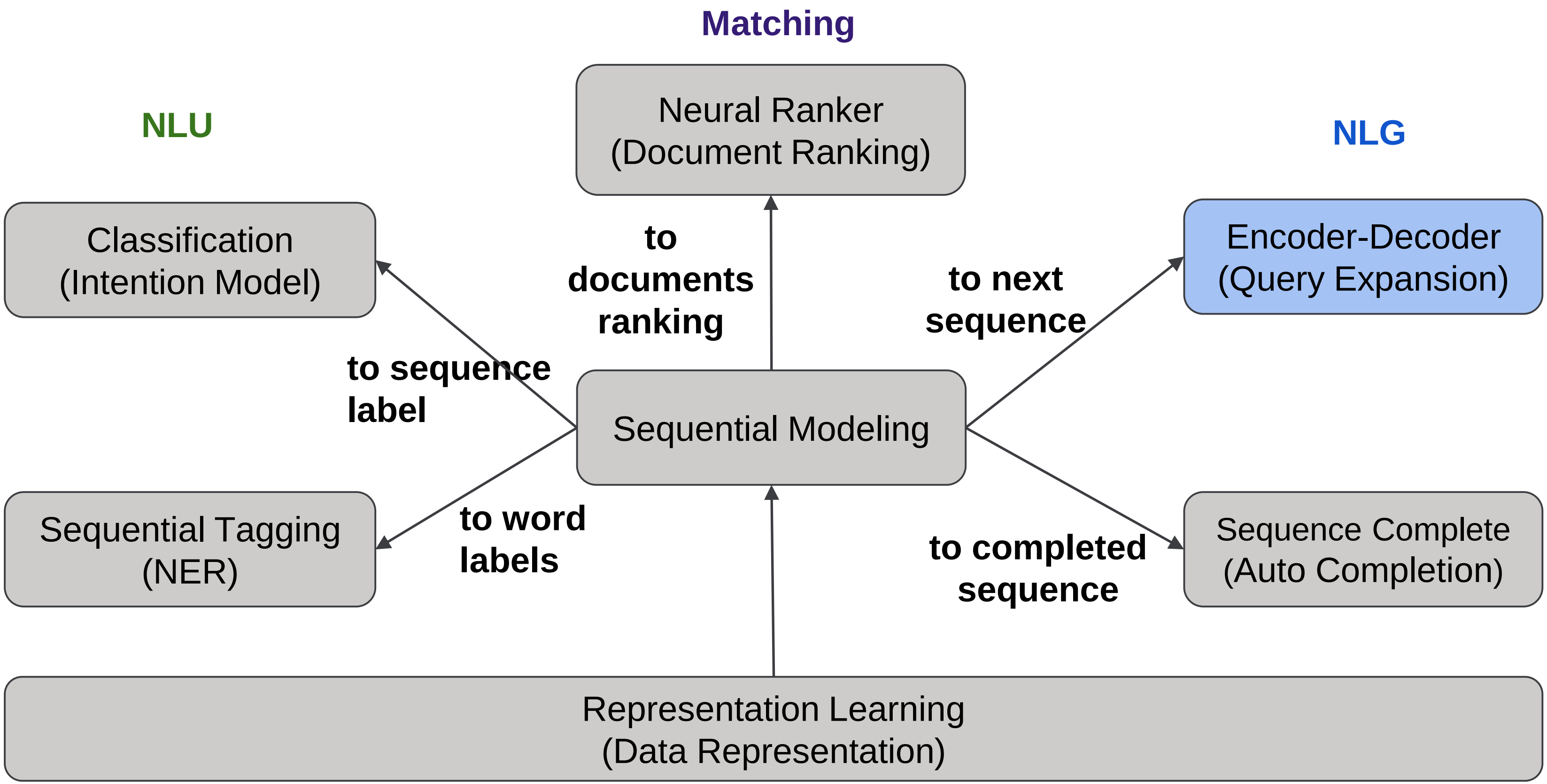


Query Tagger Performance

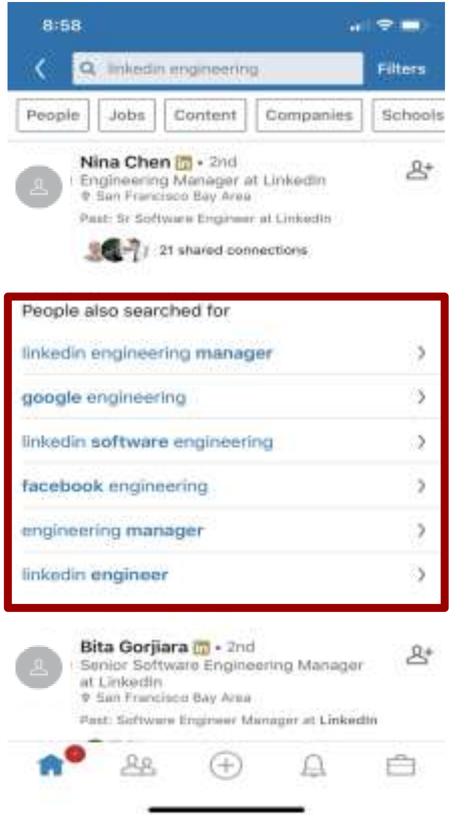


Deep & Wide Modeling Structure for Entity Tagging

Deep Learning for Natural Language Processing

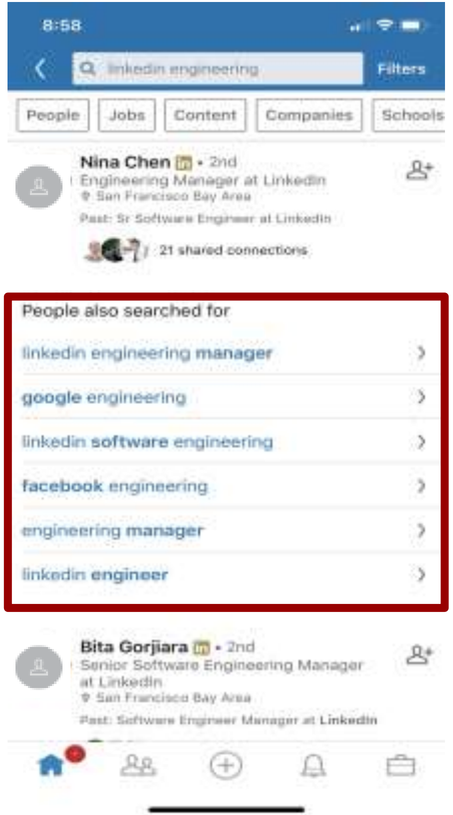


Natural Language Generation: Query Suggestion



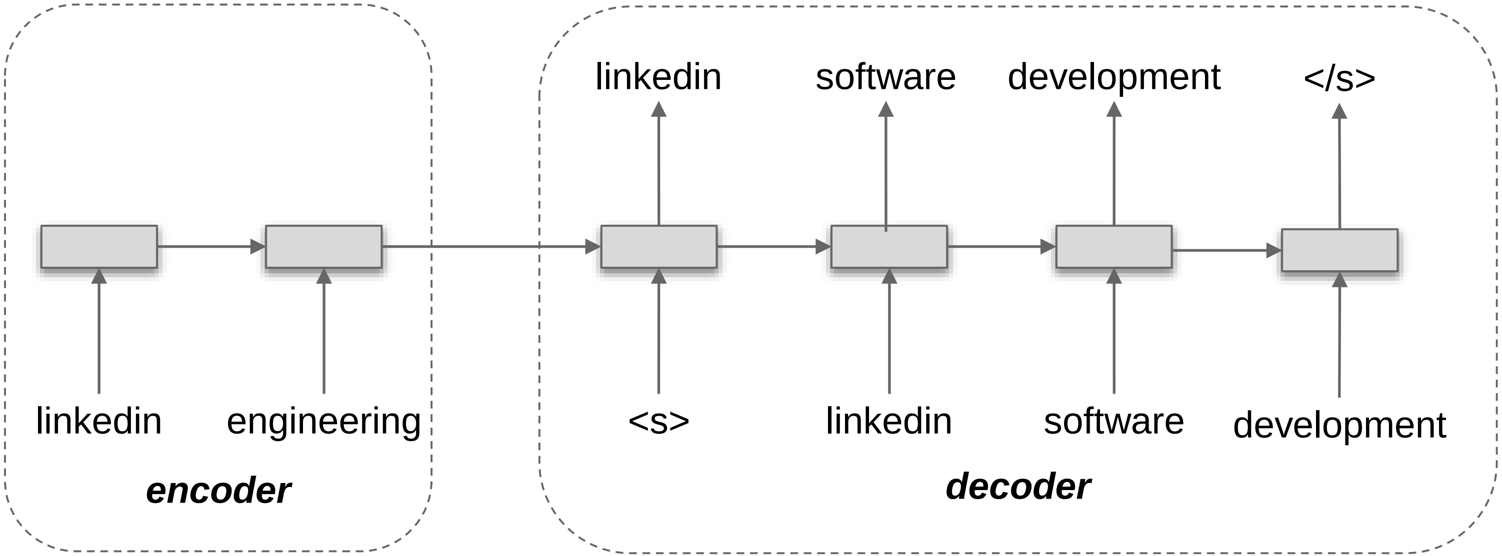
- Save User Effort of Rephrasing Informative Queries
- Capture Users' Search Intention
- Automatic Query Reformulation

Natural Language Generation: Query Suggestion



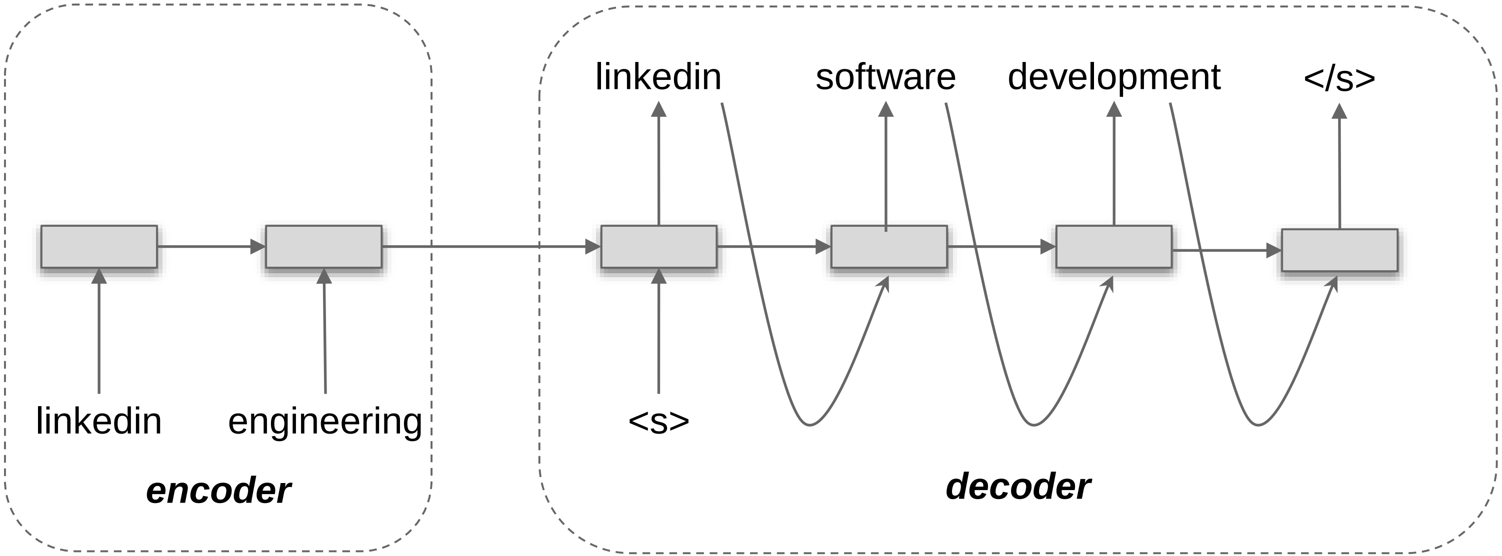
- Traditional Frequency Based Methods
  - Collect <q1, q2> pairs from search log
  - Save the frequent pairs in a key-value store
- Lack of Generalization
  - Purely string matching
  - Cannot handle unseen queries, rare words
- Seq2seq: Model Query Reformulation Behavior

Query Suggestion: Reformulate to Related Queries



- Training: the 2nd query is given
- Maximize $P(\mathbf{y}|\mathbf{x}) = \prod P(y_i|h_i)$

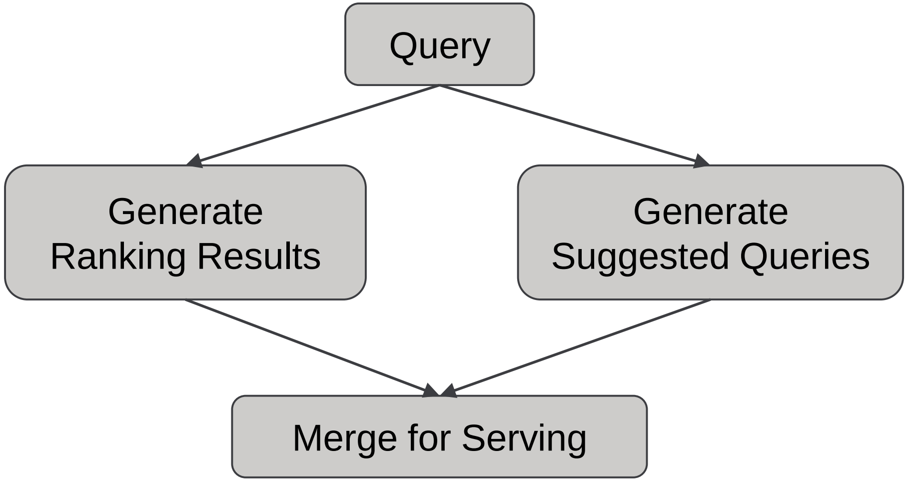
Query Suggestion: Reformulate to Related Queries



- Inference: the 2nd query is unknown
- Beam search instead of greedy search

Query Suggestion: How to Handle Online Latency

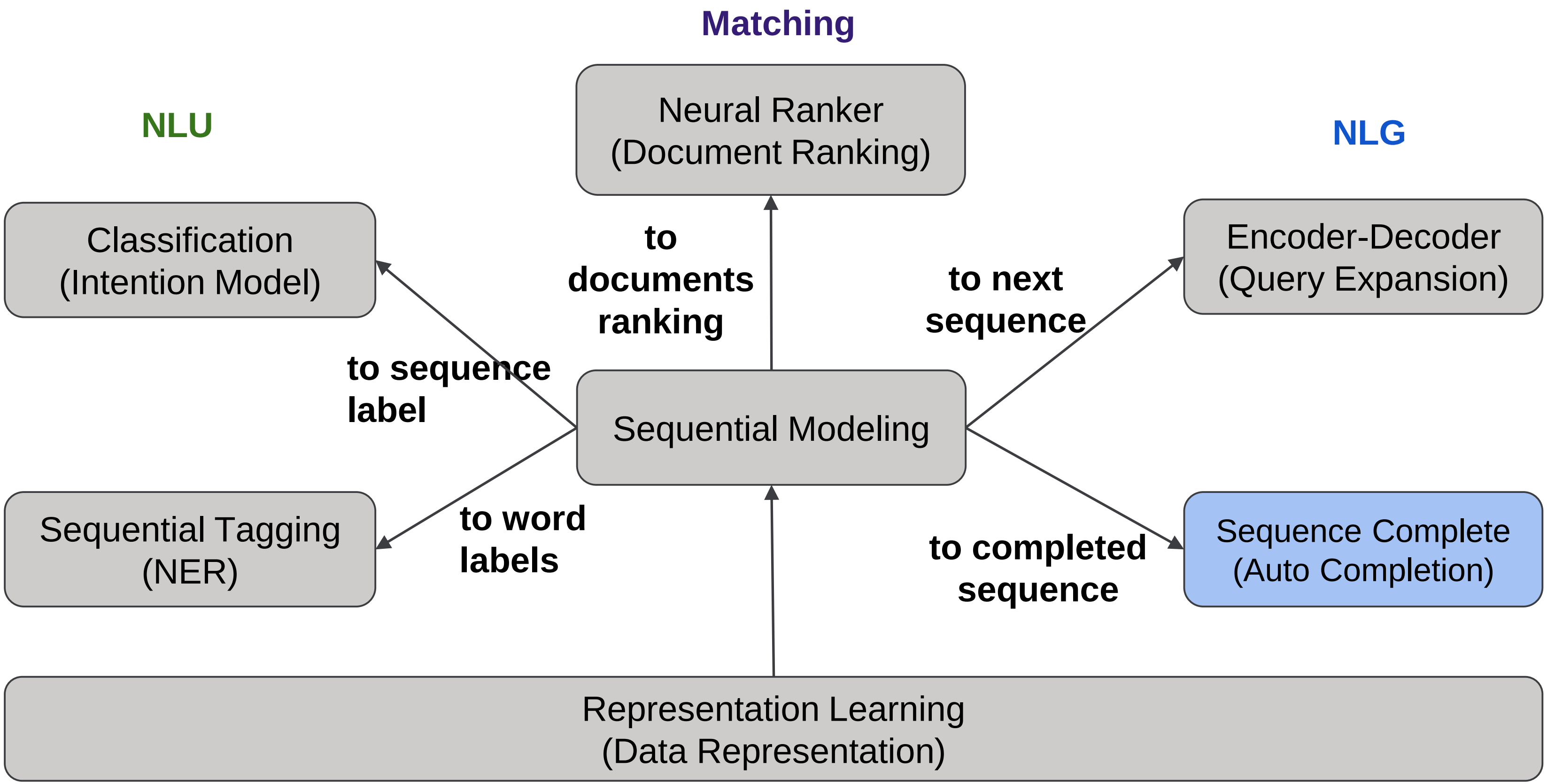
- Latency is strictly constrained for one query
  - Make it parallel with search ranking



Online Performance

- English Market
  - Coverage: +80% Impressions, +26% CTR
  - +1% Total job application
- I18n Market
  - +3.2% Successful Searches
  - +28.6% First Order Actions

Deep Learning for Natural Language Processing

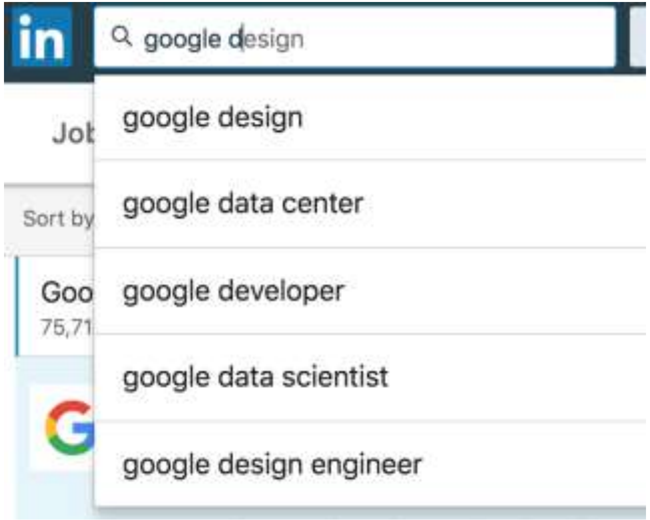


Natural Language Generation: Auto-Completion

- Given a prefix, predict the **completed query**, rather than the **completed word**

Auto-completion Challenges

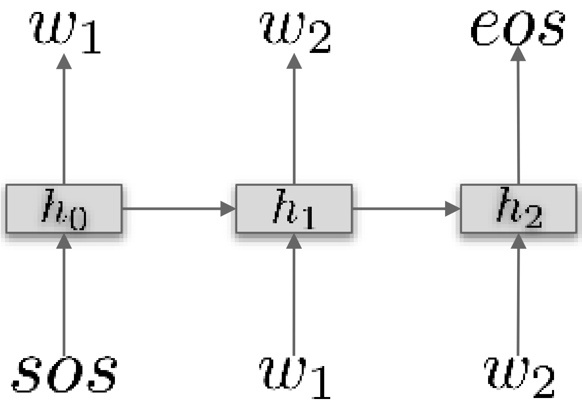
- **Short Context**
  - How to enrich semantic features
- **Personalization**
- **Latency Restriction**
  - Have to adopt simple and fast models



A Two-step Approach: Generation and Ranking

- **Candidate Generation**
  - Collect query frequency from search log
- **Candidate Ranking**
  - Neural Language Model serves as a scoring function

Auto-Completion: Neural Language Model as Scoring/Ranking



$$s(q) = \sum_i \log P(w_{i+1} | h_i)$$

- Achieved 6x speedup with optimization

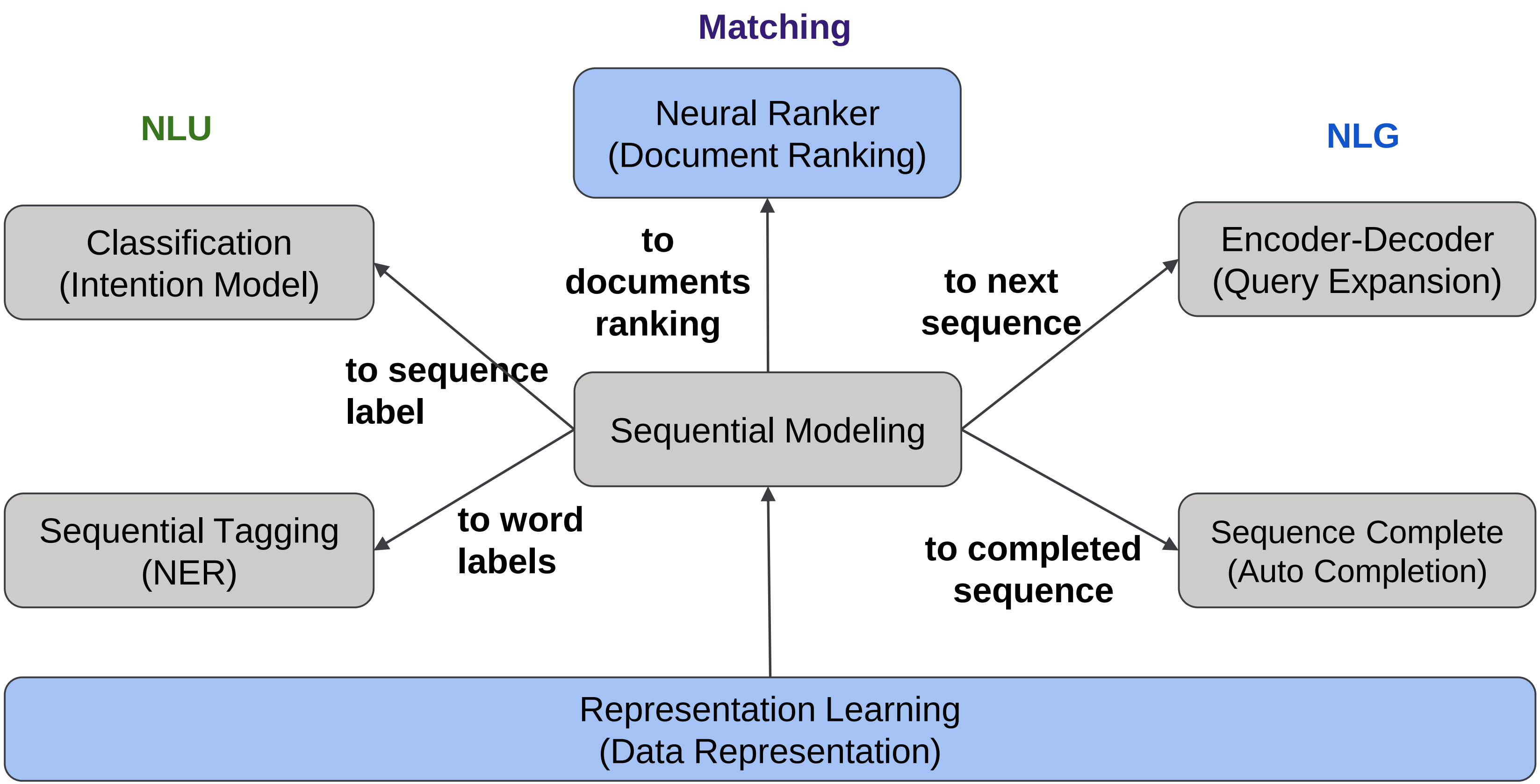
Auto-Completion Online Experiments

Neural Auto-completion vs Frequency Based Methods

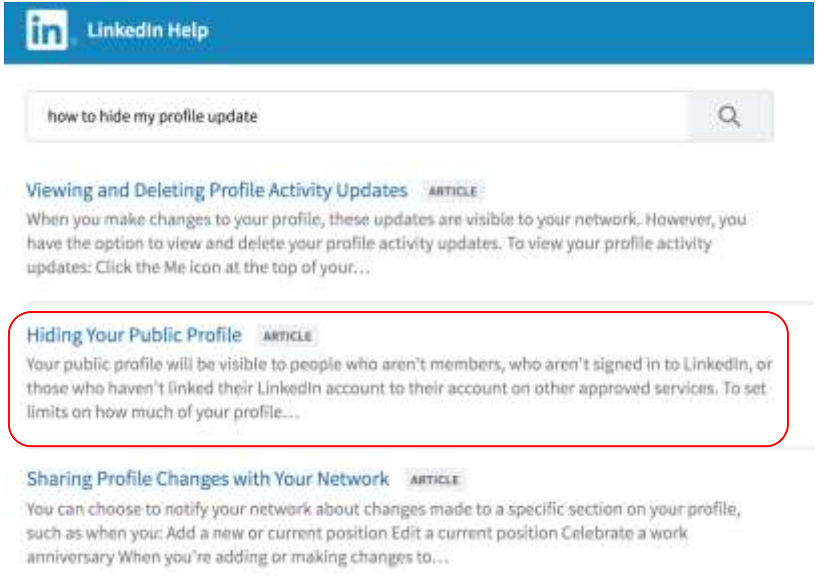
- People Search English Market
  - +88.75% autocomplete clicks
  - +0.81% CTR@1 in SERP

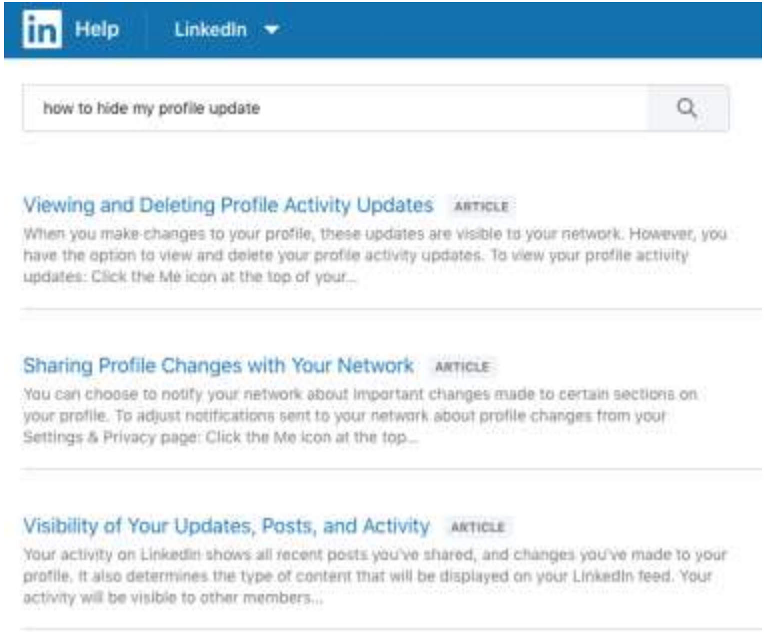
- Job Search German Market
  - +3.24% Results clicked
  - -6.24% Raw searches w/o results
  - +3.24% Entity view

Deep Learning for Natural Language Processing

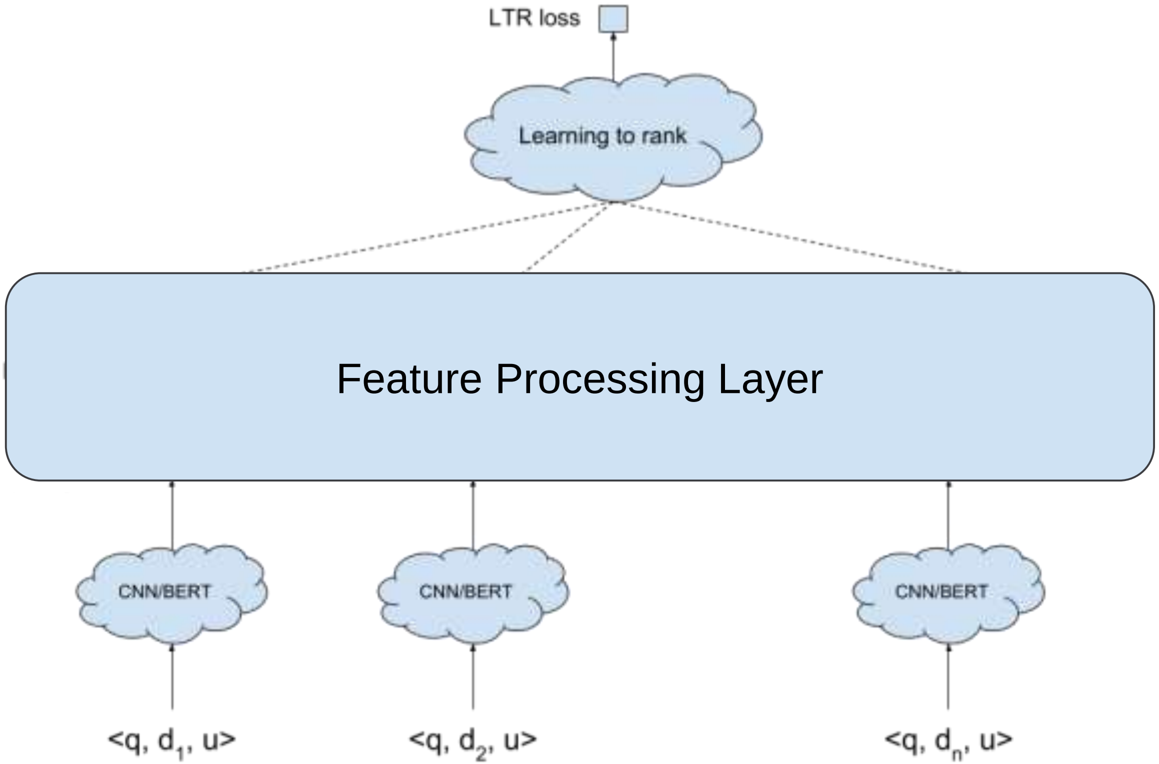


Natural Language Processing: Document Ranking





Neural Ranking



Experiments

Offline

People Search Ranking (NDCG@10)	
Wide Features	
CNN	+1.32%
BERT (google pretrained)	+1.52%
BERT (linkedin pretrained)	+1.96%

Online

- Help center ranking (BERT vs CNN)
  - +9.5% search CTR
  - +7% search Clicks

Lessons & Future Trends

- **Ground Truth Data Availability**
  - Human Labelled Data (Crowdsourcing)
  - Behavior Data
  - Mixture Data
  - Automatic Data Generation with Generative Models (e.g., GANs)

- **Model Debugging** in Complicated Search Systems
  - Model Interpretation
  - Model Reduction

Lessons & Future Trends (Cont'd)

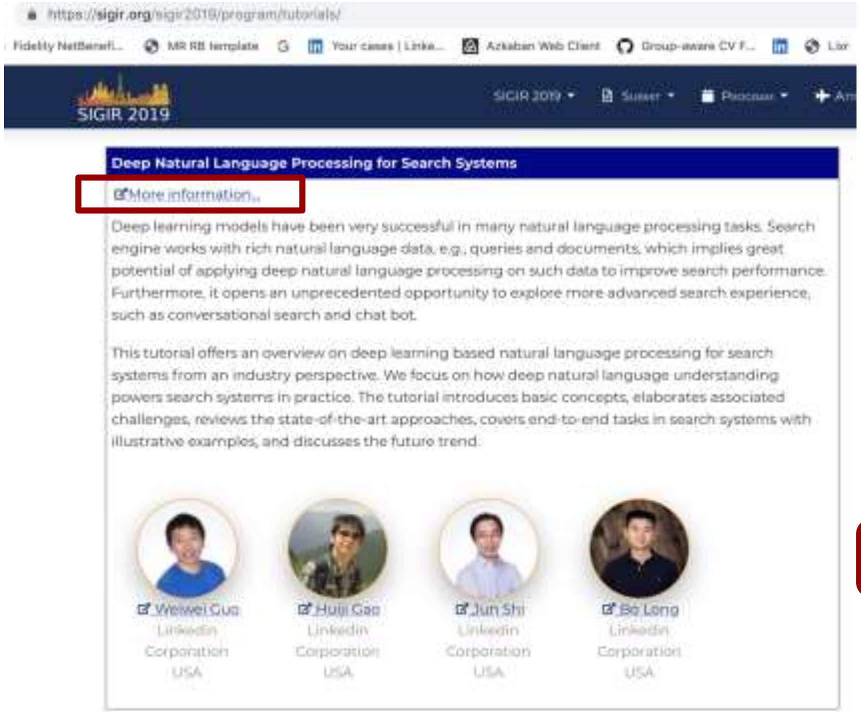
- **Efficient Modeling Training and Serving**
  - Automatic Hyperparameter Tuning & Structure Learning
  - Memory and Latency Optimization for Efficient Online Serving

- **Model Generalization**
  - Pre-trained Model for Multiple Products

- **Internationalization**
  - Transfer Learning
  - Multilingual Learning

Feedback

- **Your Feedback is Greatly Appreciated**



Slides will be Available on Website

Thank You!