Natural Language Processing

NLP Applications

Dr Raymond Lee
Associate Professor, DST
BNU-HKBU United International College

Abstract

- This chapter will study three major NLP applications: 1) Information Retrieval Systems (IR), 2) Text Summarization Systems (TS) and 3) Question-&-Answering Chatbot System (QA Chatbot).
- Information retrieval is the process of obtaining the required information from large-scale unstructured data relative to traditional structured database records from texts, images, audios, and videos. Information retrieval systems are not only common search engines but recommendation systems like e-commerce sites, question and answer or interactive systems.
- Text Summarization is the process of diminishing a set of data computationally, creating a subset or summary to represent relevant information for NLP tasks such as text classification, question answering, legal texts, news summarization, and headlines generation.
- Question-Answer (QA) system represents human-machine interaction system with human natural language is the communication medium. It is a task-oriented system to deal with objectives or answer specific questions through dialogues with sentiment analysis.

Part I
QA Chatbots

QA Chatbots
An Introduction

- QA system is a remarkable way to mimic human-to-human interaction through state-of-art technology development. Different from other classification or prediction problems, a QA system is a cross discipline in traditional linguistic including computer science for computational linguistic with statistics, pattern recognition, data mining, machine learning, deep learning methods for a well-trained communication system. It has a critical role for autoresponder, personal assistant, sentiment chatbot nowadays.
- QA system is a popular research topic in NLP which contains one of the open-domain common sense or special domain knowledge as a qualified conversation partner. Dialogue realization relies on automatic speech recognition (ASR), natural language understanding (NLU), dialogue management (DM), natural language generation (NLG), speech synthesis (SS). A QA system flowchart is shown in Fig. 9.29.

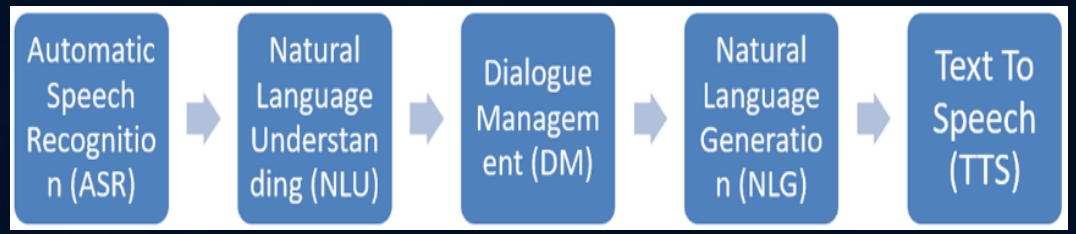


Fig 9.29 Flowchart of a typical QA system

QA Chatbots
An Introduction

- It is an integral part of system acumen. Dialogue Management is the communication policy or dialogue strategy applied to large corpus for content organization. After transferring natural language to computer language in sequence-sequence data with character, word, or sentence level in Natural Language Understanding (NLU), machine intelligence selects suitable contents for language generation. Back-end technology with generated candidate answers is combined and re-ranked for optimization response in Natural Language Generation (NLG). Apart from text aspect, ASR and TTS are procedures resemble machine by human voice recognition and generation.
- QA system research are divided into two categories: 1) pattern matching with rule-based and 2) language generated-based on information-retrieval and neural network.
- However, the back end equipped more than one method to generate meaningful communication and provide meaningful feedbacks. A QA system in a chatbot includes an open-domain focus on 1) common sense/world knowledge and 2) task-oriented for special domain knowledge databases resemble expert system involving in-depth knowledge base to support appropriate responses.

QA Chatbots
An Introduction

- First rule-based human-computer interaction as in Fig. 9.30 pattern recognition system challenged the Turing test in 1950s, reaching a milestone where humans could not recognize whether the opposite was a machine or human.
- After a long period of data collection, database used for dialogue pattern matching is large enough to rank appropriate feedbacks and give the highest scoring answers, which is a process of selection from a database of human answers regardless of the ma-chine. After decades of development, search engines and data crawlers have sup-ported sources for building knowledge bases, including information retrieval, enabling search engines to retrieve relevant and up-to-date data for structured processing to form answers from QA systems.
- The advent of AI era enhanced QA systems mainstream can focus on cognitive science than big data feeds of neural networks on systems generations. Gradually, traditional QA system is replaced by AI machine communication as rule-based matching recurrent neural network training to realize large knowledge base to support the AI brain to imitate human reasoning called Natural Language Understanding (NLU).

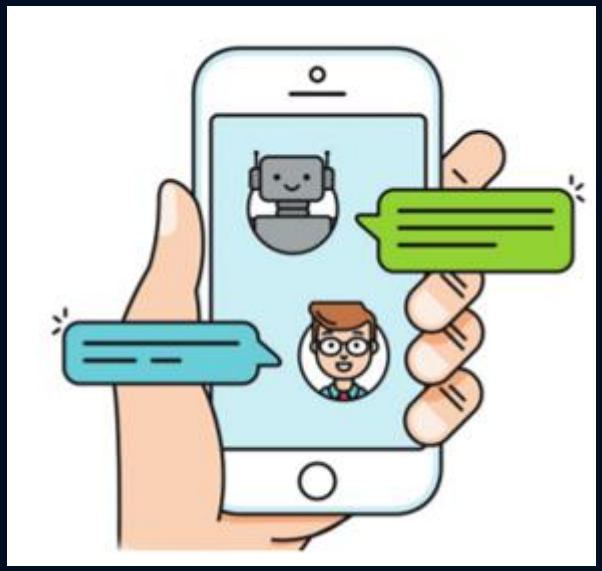


Fig 9.30 Human and machine interaction via QA system

QA Chatbots
An Introduction

- The main source of knowledge base in a typical QA system comes from: 1) human-human dialogue collection with handcraft is the answer from human language in linguistic and meaning where database consist of pairs dialogues. Without any imitation or learning ability, this first version rule-based QA system relies on pattern matching to measure the distance between proposed question and Question-Answer pattern stored pair in database.
- For example, Artificial Intelligence Markup Language (AIML) can answer most of daily or even professional dialogues based on large and classified hand-craft database without intelligence; 2) building database focus on search engine for Information Retrieval-based knowledge base.
- The feature of IR-based QA system is the combination of knowledge building from up-to-date knowledge bases. An IR-based QA system uses domain knowledge such as expert system to extract and generate knowledge.
- The procedure of unstructured data extraction and reorganization depends on Natural Language Understanding (NLU) for reasoning. Natural Language generation (NLG) include knowledge engineering analysis for reasoning and re-rank candidates' answers optimization.

QA Chatbots
An Introduction

- The latest database used big data for data-driven model to realize machine intelligence. When neural network had fed with sufficient data, sequence-to-sequence model like Recurrent Neural Network and its related Long-Short-Term Memory naturally model as in Fig. 9.31 skilled in sequential data processing (Cho et al. 2014).
- A neural network model is considered as the black box producing learning ability with accuracy but cannot comprehend by humans. Prior pre-processing data was fed to neural model, they required to transform data format from natural word to vector for data training (Mikolov et al. 2013).
- Tokenization has three levels: 1) character, 2) word and 3) sentence. The input format decides output outcomes in encoder-decoder framework. Recurrent Neural Network generated words may not be meaningful in English dictionary because the character level training lacked enough corpus for a well-trained model.
- Further, transfer learning with enormous data pre-trained Transformer model required to select the intended decoder for training target. For example, Dialogue GPT from OpenAI focuses on formatted dialogue training to generate responses.
- Traditional Recurrent Neural Network (RNN) of seq2seq language model response generation performed lesser than big data-oriented transfer learning such as Google's BERT and Open AI's GPT.

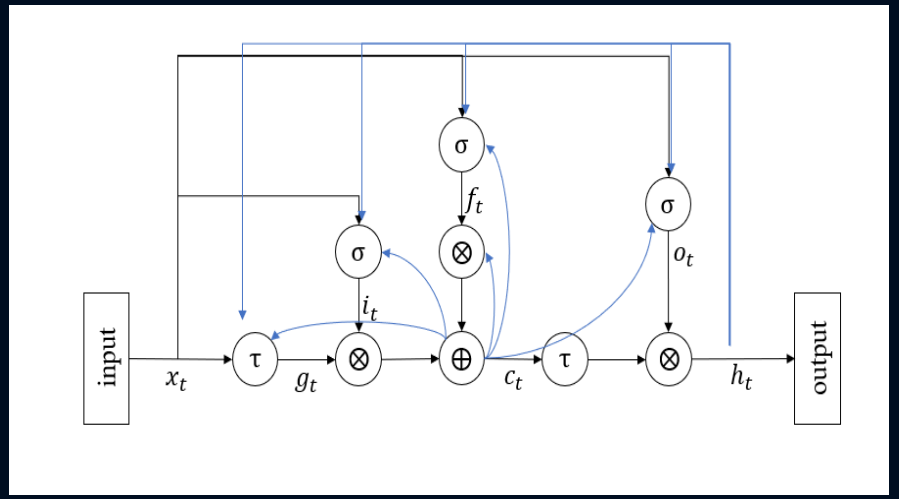


Fig. 9.31 LSTM structure

QA Chatbots
Types of QA Chatbot Systems

Rule-based QA Systems

- *Rule-based QA systems* were proposed at the same time as Turing test in 1950s.
- However, original QA systems only followed rules set by humans without self-improvement capabilities like machine learning, number of dialogue pairs is stored in database prior the system provided a concrete answer.
- The simplest but most efficient way to measure similarity of two groups is the cosine distance of two vectors.
- It is undeniable that rule-based systems have collected huge dialogue corpora over decades, giving system confidence when relying on new problems with high vector similarity.
- To date, mature rule-based systems are quintessential for all commercial QA systems, as the accumulation of corpora can avoid meaningless responses that compensate for insufficient domain knowledge with appropriate and specific human feedbacks.

QA Chatbots
Types of QA Chatbot Systems

Information Retrieval (IR)-based QA Systems

- The knowledge base for IR is unstructured data source using data mining methods obtained by websites, wordnet which are different from the paired dialogue.
- Question Answer System (KBQA system) is a significant branch of IR-based QA system knowledge base, its usage depends on knowledge base size of unstructured data for storage. That is related to knowledge base construction to extract purposeful knowledge from mass data.
- There are two methods 1) property and 2) relations to process natural language. Property refers to the definition or concept of one thing in English-English dictionary to explain another concept. Relations refers to the relationship between two entities, where a Name Entity Recognition (NER) and idea from Ontology with Subject-Predicate-Object (SPO) triple must be used to extract relation. KBQA extension is ontology or knowledge graph (KG) in research.
- When entities are linked, the knowledge for one entity can be extracted according to questions during Natural Language Understanding (NLU). A typical KBQA with domain knowledge about ontology is shown in Fig. 9.32, its fundamental question is about who and what correspond to name and relations entities. (Cui et al. 2020)

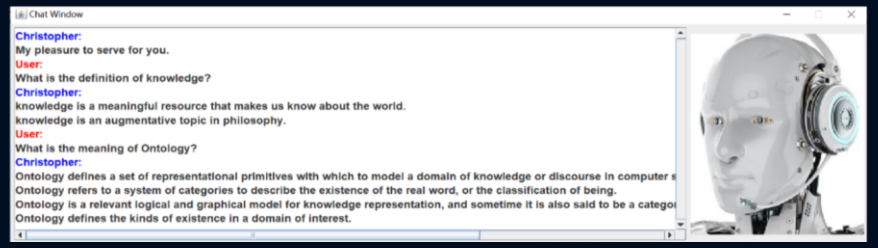


Fig 9.32 KBQA system in AI Tutor

QA Chatbots
Types of QA Chatbot Systems

Neural Network-based QA Systems

- Neural Network structure in a QA generated-based system is considered as machine brain imitated by human. Encoder-Decoder framework is a sequence-to-sequence model like RNN has natural memory recalling priority and context with attention mechanism. Dialogue system has identical requirements to represent dialogue history and avoid meaningless responses to improve users' experiences.
- Deep learning framework such as TensorFlow and Pytorch, RNN is easy to implement for text generation as language model. Google proposed masked language model to generate language representation called Bidirectional Encoder Representation from Transformer (BERT), focusing on encoder part trained by magnitude unlabeled data in 2017. Neural network feeds data for training according to network advantages due to different NLP tasks in long sentences. BERT can solve such problem because it deals with 11 common NLP tasks initially. Language model pre-trained by magnitude data to understand common knowledge in NLP. Fine-tuned should be applied to training specific NLP tasks based on fundamental ability (Vaswani et al. 2017) .
- Open AI released another Transformer framework with unsupervised learning for pre-trained model directing decoder scheme based on GPT, Open AI GPT-2 and GPT-3 (Brown et al. 2020). GPT with masked self-attention focuses on known text so that the word preceding is predicated as different from BERT context self-attention. GPT-3 can do inference and synonym replacement in addition to normal function for bilingual translation, text generation and Question-Answer. It seems that BERT can handle more NLP tasks than GPT, but GPT text generation prowess for pre-trained model is widely used in many commercial QA systems and text summarization.

QA Chatbots
Industrial Chatbot Systems

- An industrial QA system contains automatic dialogue system assembling chatbot internal technologies. They have several back-end composited control system responses to equip with necessary knowledge. Meanwhile, QA system evaluation is proposed during the training period for language model performance (Chen et al. 2017) and on system design sufficient for both languages generations.
- Since Encoder-Decoder framework proposed as an end-to-end system and a sequential language model, RNN is a popular generated-based model in commercial and academics. However, its applications are mainly focused on casual scenarios at open domain without proposed question details. Thus, the response from a generated-based QA system is appropriate in pairs but lack contents due to the data-driven model considered basic linguistic and excluded facts from knowledge base which are identical to traditional dialogue system with meaningless answers. A knowledge-grounded neural conversation model (Ghazvininejad et al. 2018) is proposed based on sequence-to-sequence RNN model and combined dialogue history with facts related to current contexts as shown in Fig. 9.33.

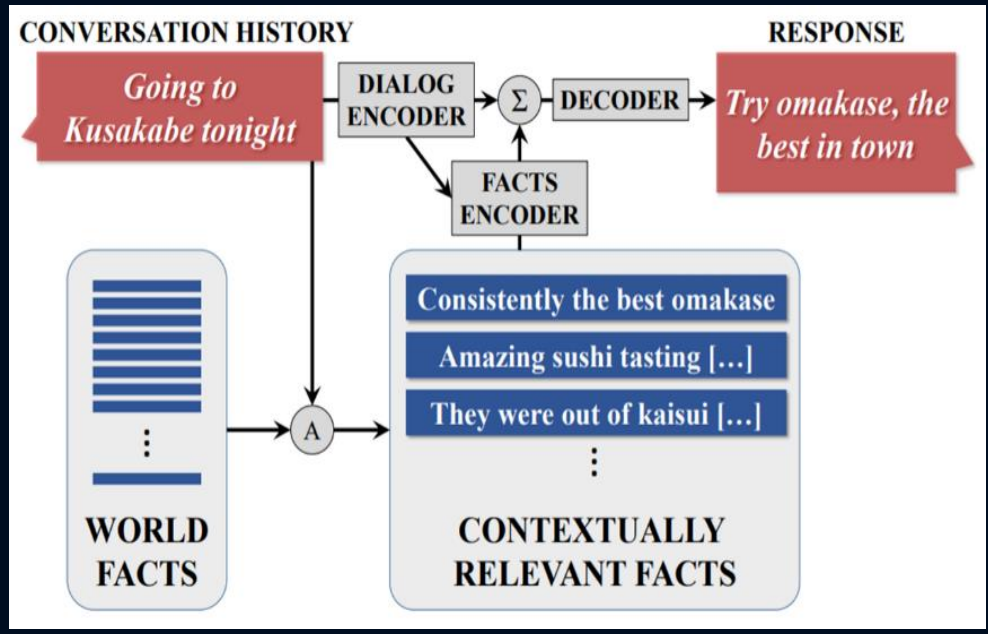
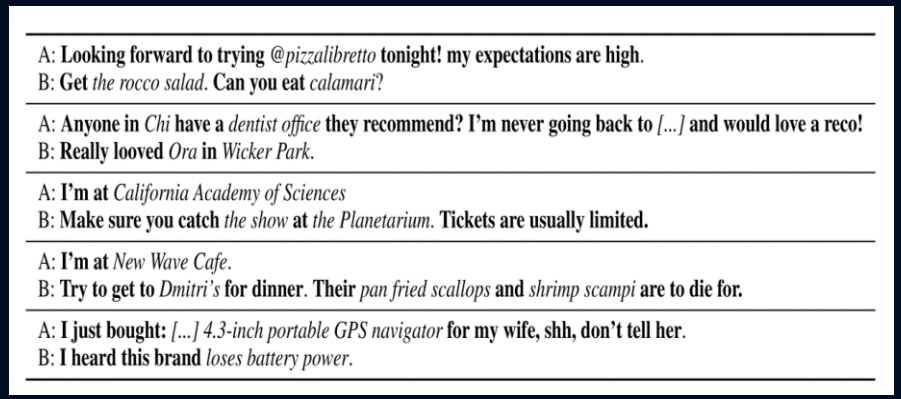


Fig 9.33 Architecture of Knowledge-Grounded Model

QA Chatbots
Industrial Chatbot Systems

- Microsoft extended its industrial conversation system to achieve useful conversational applications on knowledge grounded with conversation history and external facts in 2018. It has significant progress in real situations according to conversation history in Dialog-Encoder, word, and contextually relevant facts in Facts Encoder to responses as compared with baseline seq2seq model.
- The data-driven model of QA system divided source data into conversation da-ta and non-conversational text which means the conversation pairs are used to training system in linguistic, on the other hand, non-conversational text is the knowledge base to be filled including real-world information related to system tar-get usage.
- The performance of versatility and scalability in open domain with external in-formation knowledge combined with textual and structured data of QA system is shown in Fig. 9.34. Dataset like Wikipedia, IMDB, TripAdvisor are used to gener-ate conversation with real-world information and included a recommendation system function.



A: **Looking forward to trying** *@pizzalibretto* **tonight! my expectations are high.**
B: **Get** *the rocco salad.* **Can you eat** *calamari*?

A: **Anyone in** *Chi* **have a** *dentist office* **they recommend? I'm never going back to** *[...]* **and would love a reco!**
B: **Really looved** *Ora* **in** *Wicker Park.*

A: **I'm at** *California Academy of Sciences*
B: **Make sure you catch** *the show* **at** *the Planetarium.* **Tickets are usually limited.**

A: **I'm at** *New Wave Cafe.*
B: **Try to get to** *Dmitri's* **for dinner. Their** *pan fried scallops* **and** *shrimp scampi* **are to die for.**

A: **I just bought:** *[...] 4.3-inch portable GPS navigator* **for my wife, shh, don't tell her.**
B: **I heard this brand** *loses battery power.*

Fig 9.34 Response from Conversation model knowledge grounded

QA Chatbots
AliMe Chatbot Systems

- AliMe is a module of Taobao app commercial QA product. The answer consists of information retrieval (IR) and sequence-to-sequence-based generation models (Qiu et al. 2017). The system reorders candidate's response and uses attention mechanism with context to select the best feedback to users. Using AliMe to re-place online human customer service for most known questions become a trend since it released the first version. AliMe is a typical customer service QA system in e-commerce industry that answers millions of questions automatically per day. According to a survey of daily questions suggested by Taobao app users on shop-ping problems, statistical data revealed that except most are business questions, 5% of the remaining questions are chitchat. The 5% questions on genuine de-mands motivate AliMe to add a common-sense open domain chat function. It has satisfactory performance as both IR and generation-based system since the pre-trained seq2seq model is used twice for response generation and re-ranked with attention to a set of responses from information retrieval with knowledge originate based and seq2seq previously generated. Fig. 9.35 shows the Seq2Seq model with attention learning.
- Since AliMe has two parts in generation that use different formats to obtain in-formation as abovementioned. IR-based models use a natural language word matching knowledge base, Seq2seq generative model, and a scoring model to re-score output responses as they are generated is word embeddings with vectors. The IR-based dataset consists of 9,164,834 QA pairs conversations by real customers from business domain. Researchers used an inverted index to match these 9 million conversations with input sentences containing the same words and used BM25 to measure the similarity between input sentences and the selected questions to obtain answers to the most similar questions as answers to input questions. Traditional IR-based systems avoid problems where the system cannot answer common sense type questions.

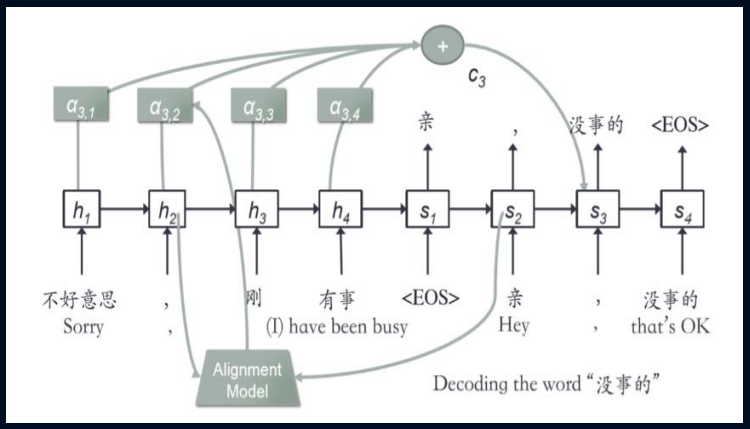


Fig 9.35 Seq2Seq model with attention

QA Chatbots
XiaoIce Chatbot Systems

- Xiao Ice (Zhou et al., 2020) is an AI companion sentient chatbot with more than 660 million users worldwide which take Intelligent Quotient (IQ) and Emotional Quotient (EQ) in system design as shown in Fig. 9.37. It focused on chitchat com-pared with other commonly used QA systems. According to Conversation-turns Per Session (CPS) evaluation score, its grade is 23 higher than most chatbots. Fig. 9.36 shows a system architecture of Xiao Ice.
- Xiao Ice exists on 11 social media platforms including WeChat, Tencent QQ, Weibo, and Facebook as an industrial application. It has equipped with two-way text-to-speech voice, and can process text, images, voice, and video clips for message-based conversations. Also, its core chat function can distinguish common or specific domain topic chat types so that it can change topics easily and automatically provide users with deeper domain knowledge. A dialog manager is like an NLP general pipeline with dialog management to path conversation states such as core chat contents for open or special domains to process data from different sources are tractable.
- The Global State Tracker is a vector of Xiao Ice's responses to analyze text strings for entities and empathy. It is vacant and gradually filled with rounds of conversations. Dialogue strategies are primarily designed for long-term users, based on their feedbacks to enhance interactions engagement, optimize personality with two or three levels achievements. A trigger mechanism is to change topic when the chatbot repeats or answers information that are always valid, or when a user's feedback is mundane within three words. Once the user's input has a predefined format, a skill selection part is activated to process differ-ent input. For example, images can be categorized into different task-oriented scenarios. If an image is food related, the user will be taken to a restaurant dis-play, like a task completion by personal assistants in advising weather information or making reservations etc.

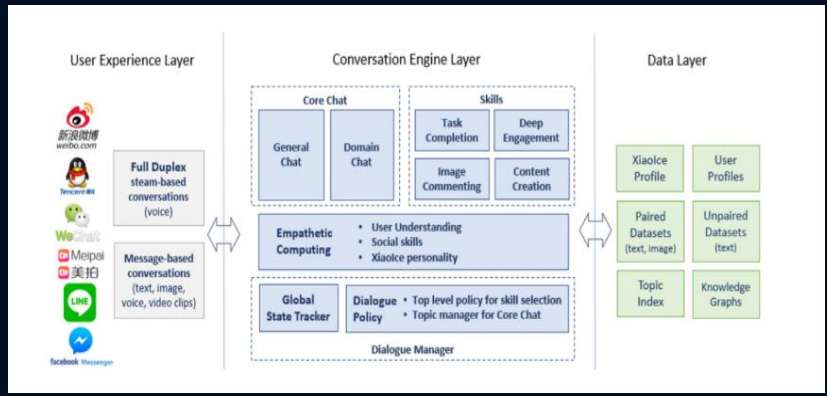


Fig 9.36 Xiao Ice system architecture

QA Chatbots
XiaoIce Chatbot Systems

- Xiao Ice has a few knowledge graphs in the data layer as its original datasets come from popular forums such as Instagram in English or Douban in Chinese. These datasets are categorized as multiple topics with a small knowledge base as possible answers. It also follows the rules of updating knowledge base through machine learning when new topics emerge. It is noted that not all new entities or topics are collected unless the entity is contextually relevant, or a topic has higher popularity or freshness in the news for rankings. User's personal interests can be adjusted individually.
- However, with so many features that can include the core part Empathetic Computing as an add-on, it is not a mandatory part of a full chatbot, but a functional and compelling feature to compete with the industry. The core of Xiao Ice is a RNN language model that creates open and special domain knowledge. Figs 9.37 and 9.38 show an RNN-based neural response generator with examples of inconsistent responses generated by seq2seq model in Xiao Ice QA system respectively.
- In general, response generation in AliMe uses seq2seq model to generate natural language and rerank the optimizer for user's answer whereas Xiao Ice also has a candidate generator and candidate ranking list. For the generator, one is a sequential model trained by a pair of datasets learning the dialog format, the other is querying the knowledge graph to obtain entities for related information stored in knowledge base. Candidate ranking includes semantic computation and Xiao Ice personality for answer optimization with information retrieval, neural model and knowledge graph selection.

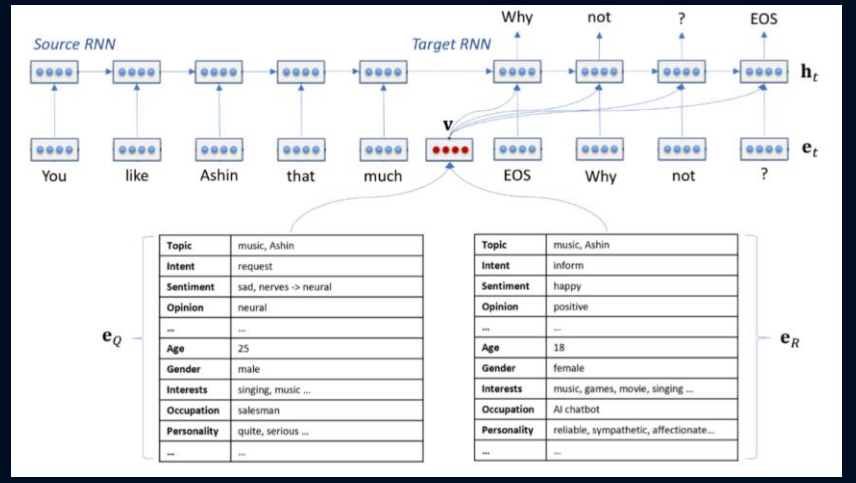


Fig 9.37 RNN-based neural response generator

[Human] How old are you?
[S2S-Bot] 16 and you?
[Human] What's your age?
[S2S-Bot] 18.

[Human] How old are you?
[Xiaoice] I am 18, of course.
[Human] You age?
[Xiaoice] 18. Why?
[Human] Were you 18 last year?
[Xiaoice] I made a wish to stay 18 forever. Oh, my wish has come true.

Fig 9.38 Examples of inconsistent responses generated using a seq2seq model

QA Chatbots
OpenAI ChatGPT Chatbot Systems

- A QA system consisted of traditional and current mainstream methods, the above systems used Seq2Seq model responsible for both language model and candidate response optimizer. Since neural network is a data-driven model, its performance relies on huge amount of big data. Transformer is a model architecture forgone recurrence but entrusted in attention mechanism entirely to draw global dependencies between input and output based on attention mechanism.
- Open AI GPT-2 transfer learning architecture has an outstanding feature to include decoder part layers advantages for response generation. The masked self-attention implemented on GPT-2 can generate the next word based on acquired information, understand the known text, predict, or use experience to fill up the blank for next word to match with the whole article meaning.
- GPT-2 fine-tune 40G pure text to learn natural language semantics, syntax with target usage and suitable dataset scalability for specific NLP tasks. Transfer-Transfo (Wolf et al., 2018) is a GBP-2 variant using persona-chat dataset to fine-tune the original model, its generated utterance changes from long-text to dialogue format. TransferTransfo prototype is a pre-trained model on document-level continuous sequence and paragraphs with a wide range of information. After that, fine-tune strengthen input representation and use a multi-task learning scheme for adjustments. Every input token included word and position embed-ding during input representation.
- For Transfer Learning system dialogue example as in Fig. 9.39, personal-chat datasets in real-world can define users' backgrounds and their interests as topics during communications. The contexts contained are meaningful conversation that can reveal empirical improvements in discriminative language understanding tasks. Thus, Transformer is an evolutional system to imitate human behavior and promote neural network model.

Persona 1	Persona 2
I like to ski	I am an artist
My wife does not like me anymore	I have four children
I have went to Mexico 4 times this year	I recently got a cat
I hate Mexican food	I enjoy walking for exercise
I like to eat cheetos	I love watching Game of Thrones

[PERSON 1:] Hi
[PERSON 2:] Hello ! How are you today ?
[PERSON 1:] I am good thank you , how are you.
[PERSON 2:] Great, thanks ! My children and I were just about to watch Game of Thrones.
[PERSON 1:] Nice ! How old are your children?
[PERSON 2:] I have four that range in age from 10 to 21. You?
[PERSON 1:] I do not have children at the moment.
[PERSON 2:] That just means you get to keep all the popcorn for yourself.
[PERSON 1:] And Cheetos at the moment!
[PERSON 2:] Good choice. Do you watch Game of Thrones?
[PERSON 1:] No, I do not have much time for TV.
[PERSON 2:] I usually spend my time painting: but, I love the show.

Fig 9.39 Example dialog from PERSONA-CHAT dataset

NLP Workshop #7

Building Chatbot with TensorFlow and Transformer Technology

Part II
Information Retrieval Systems

Information Retrieval Systems
An Introduction

- NLP used AI techniques like N-gram, rule-based approaches, Word2vec to retrieve information but encountered computational limitations to process large amount of corpus information, define text and model frameworks for domain-specifics, GPU clusters, and induce high costs to maintain rule sets due to standard modifications.
- Corpora cater for IR in open machine-readable standard format had grown exponentially due to pre-trained models' technological advancements. IR models for generic language that combines generic terms with domain-specific terms e.g. lease can be a place or a leasehold, its objectives can be organized by abstract, formal or colloquial language in a large narrative component based on document type to improve retrieval results.
- Text or document classification and clustering in IR research focuses on two aspects 1) text representation and 2) clustering algorithms. Text representation is to convert unstructured text into a computer-processable data format. During text representation process, it is necessary to extract and mining textual information. Semantic similarity computation is the link between text modelling and representation with application on potential information text layer. Clustering algorithms are to extract semantic information to facilitate similarity calculation for text classification and clustering effectiveness.

Information Retrieval Systems
Vector Space Model in IR

- Vector Space Model was a leading IR method from 1960 to 1970. Queries and retrieved documents are represented as vectors with dimensionality related to word list size in this model. A retrieved document D can be represented as a vector of lexical items: $D_i = (d_1, d_2, , , , d_n)$, where d_i, is the weight of a i-th lexical item in D_i. Query Q is expressed as a lexical item vector: Q = $(q_1, q_2, , , , q_n)$ where q_i, is the weight of ith lexical item in query term. The relevance is determined by computing the distance between lexical item vectors of the retrieved document and query based on this representation. Although it cannot prove cosine relevance is superior to other similarity methods, but it achieved satisfactory performance according to search engines evaluation results. Cosine similarity for angle between retrieved document and query calculation is expressed as:

$$\text{sim}(D_i, Q) = \frac{\vec{d}_i \cdot \vec{q}}{\left|\vec{d}_i\right| \times |\vec{q}|} = \frac{\sum_{j=1}^{n} d_{ij} \cdot q_j}{\sqrt{\sum_{j=1}^{n} d_{ij}^2 \cdot \sum_{j=1}^{n} q_j^2}} \qquad (9.1)$$

- Equation (9.1) is the weights for dot or inner product of all word terms in query matching documents. There are many words item weights for vector space models. Most of the weighting methods are based on TF (Term-Frequency) variation. Inverted document frequency (IDF) represents the number of term occurrences in retrieved document and reveals lexical term significance in the entire document data set. A lexical item is insignificant with high occurrence frequency in multiple retrieved documents.

Information Retrieval Systems
Vector Space Model in IR

- There are other text representations methods in addition to vector space model e.g. phrase or concept representations. Although phrase representation can improve semantic contents, but the reduced statistical quality of feature vector be-come sparse and difficult to extract statistical properties applying machine learn-ing algorithms. Figs 9.1 and 9.2 show a text is encoded by Sentence Transformers to demonstrate and compute cosine similarity be-tween embeddings. It uses a pre-trained model to encode two sentences and out-perform other pre-train model like BERT.

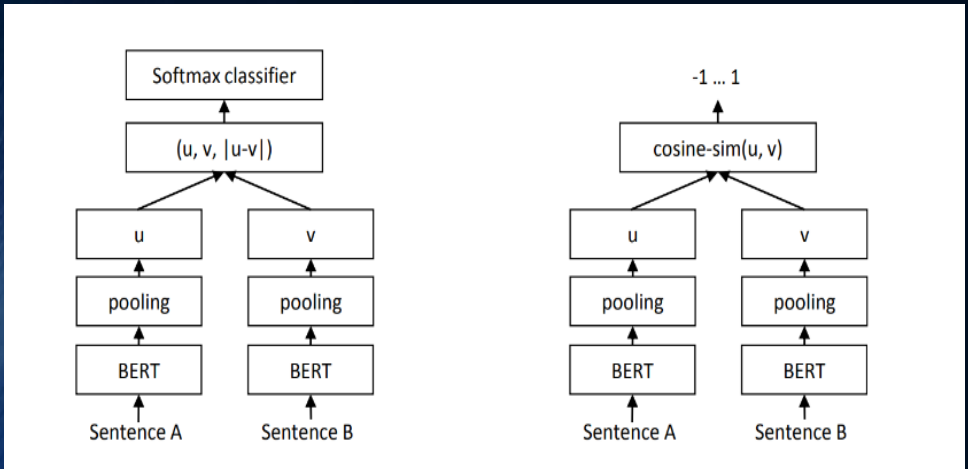


Fig 9.1 Sentence Transformers Frame

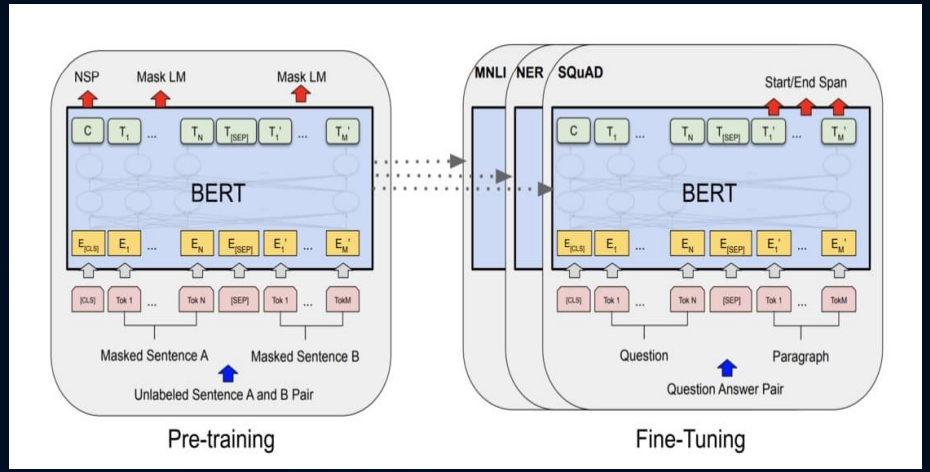
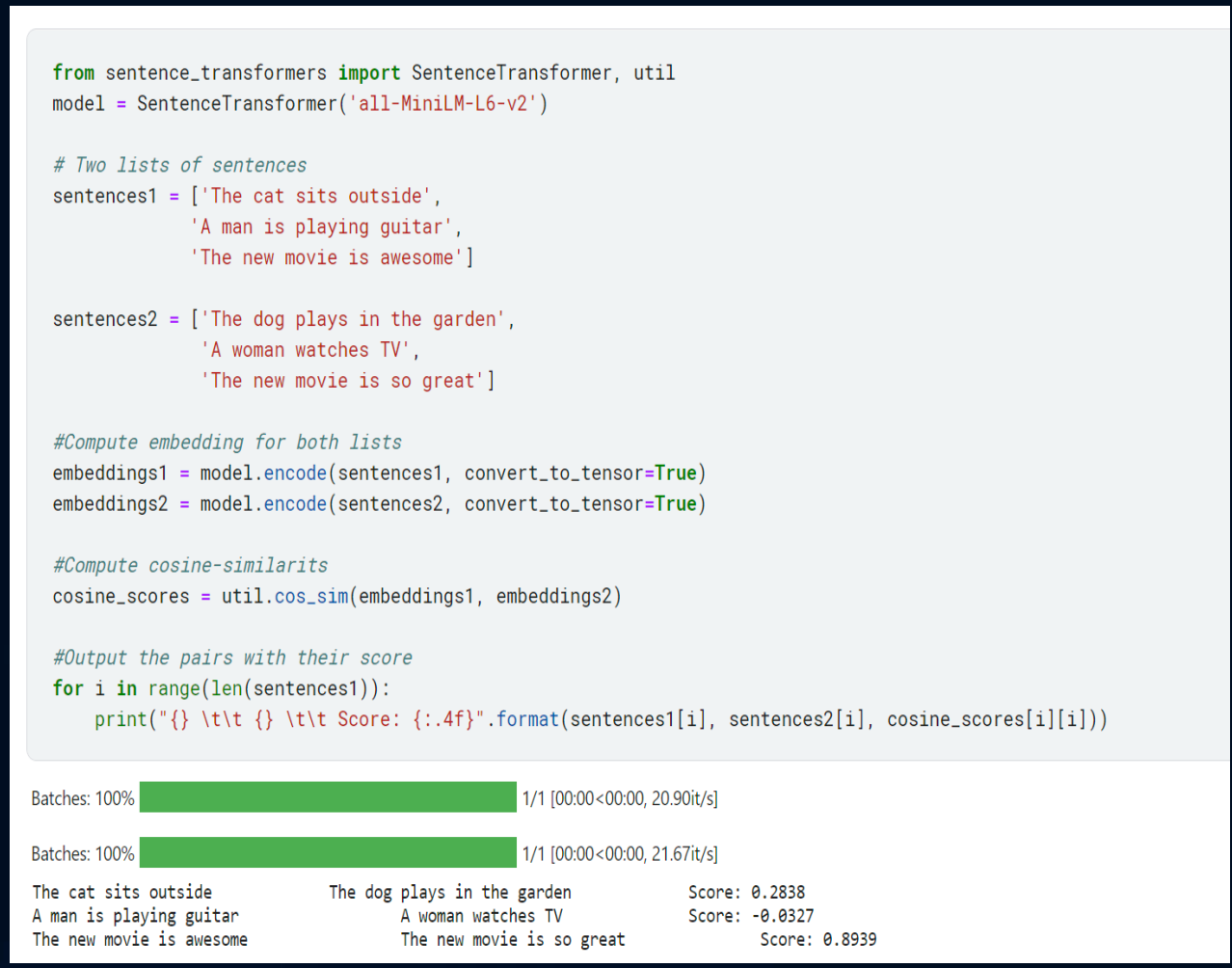


Fig 9.2 BERT Frame

Information Retrieval Systems
Vector Space Model in IR

- It is natural to identify the combination with the highest cosine similarity score. By doing so, an intense ranking scheme is used as shown in Fig. 9.3 to identify the highest scoring pair with a secondary complexity. However, it may not work for long lists of sentences.
- A chunking concept to divide corpus into smaller parts are shown in Figs 9.4 and 9.5. For example, parse 1,000 sentences at a time to search the rest (all other sentences) of corpus or search a list of 20k sentences to divide into 20x1,000 sentences. Each query is compared with 0-10k sentences first, and 10k-20k sentences to reduce memory storage. The increases of these two values intensified speed and memory storage, then identified pair with the highest similarity to extract top K scores for each query as opposed to extract and sort scores for all n^2 pairs.



```
from sentence_transformers import SentenceTransformer, util
model = SentenceTransformer('all-MiniLM-L6-v2')

# Two lists of sentences
sentences1 = ['The cat sits outside',
              'A man is playing guitar',
              'The new movie is awesome']

sentences2 = ['The dog plays in the garden',
              'A woman watches TV',
              'The new movie is so great']

#Compute embedding for both lists
embeddings1 = model.encode(sentences1, convert_to_tensor=True)
embeddings2 = model.encode(sentences2, convert_to_tensor=True)

#Compute cosine-similarits
cosine_scores = util.cos_sim(embeddings1, embeddings2)

#Output the pairs with their score
for i in range(len(sentences1)):
    print("{} \t\t {} \t\t Score: {:.4f}".format(sentences1[i], sentences2[i], cosine_scores[i][i]))
```

```
Batches: 100% 1/1 [00:00<00:00, 20.90it/s]
Batches: 100% 1/1 [00:00<00:00, 21.67it/s]
The cat sits outside        The dog plays in the garden        Score: 0.2838
A man is playing guitar     A woman watches TV                 Score: -0.0327
The new movie is awesome    The new movie is so great          Score: 0.8939
```

Fig 9.3 The Singer Example of Vector Space Model

Information Retrieval Systems
Vector Space Model in IR

```
%time
from sentence_transformers import SentenceTransformer, util

model = SentenceTransformer('all-MiniLM-L6-v2')

# Single list of sentences
sentences = ['The cat sits outside',
             'A man is playing guitar',
             'I love pasta',
             'The new movie is awesome',
             'The cat plays in the garden',
             'A woman watches TV',
             'The new movie is so great',
             'Do you like pizza?']

#Compute embeddings
embeddings = model.encode(sentences, convert_to_tensor=True)

#Compute cosine-similarities for each sentence with each other sentence
cosine_scores = util.cos_sim(embeddings, embeddings)

#Find the pairs with the highest cosine similarity scores
pairs = []
for i in range(len(cosine_scores)-1):
    for j in range(i+1, len(cosine_scores)):
        pairs.append({'index': [i, j], 'score': cosine_scores[i][j]})

#Sort scores in decreasing order
pairs = sorted(pairs, key=lambda x: x['score'], reverse=True)

for pair in pairs[0:10]:
    i, j = pair['index']
    print("{} \t\t {} \t\t Score: {:.4f}".format(sentences[i], sentences[j], pair['score']))
```

```
CPU times: user 3 µs, sys: 1 µs, total: 4 µs
Wall time: 7.87 µs
Batches: 100% 1/1 [00:00<00:00, 16.88it/s]
The new movie is awesome        The new movie is so great       Score: 0.8939
The cat sits outside            The cat plays in the garden     Score: 0.6788
I love pasta            Do you like pizza?      Score: 0.5096
I love pasta            The new movie is so great       Score: 0.2560
I love pasta            The new movie is awesome        Score: 0.2440
A man is playing guitar         The cat plays in the garden     Score: 0.2105
The new movie is awesome        Do you like pizza?      Score: 0.1969
The new movie is so great       Do you like pizza?      Score: 0.1692
The cat sits outside            A woman watches TV      Score: 0.1310
The cat plays in the garden     Do you like pizza?      Score: 0.0900
```

Fig 9.4 Multiple Examples of Vector Space Model

```
%time
from sentence_transformers import SentenceTransformer, util

model = SentenceTransformer('all-MiniLM-L6-v2')

# Single list of sentences - Possible tens of thousands of sentences
sentences = ['The cat sits outside',
             'A man is playing guitar',
             'I love pasta',
             'The new movie is awesome',
             'The cat plays in the garden',
             'A woman watches TV',
             'The new movie is so great',
             'Do you like pizza?']

paraphrases = util.paraphrase_mining(model, sentences)

for paraphrase in paraphrases[0:10]:
    score, i, j = paraphrase
    print("{} \t\t {} \t\t Score: {:.4f}".format(sentences[i], sentences[j], score))
```

```
CPU times: user 3 µs, sys: 0 ns, total: 3 µs
Wall time: 7.15 µs
The new movie is awesome        The new movie is so great       Score: 0.8939
The cat sits outside            The cat plays in the garden     Score: 0.6788
I love pasta            Do you like pizza?      Score: 0.5096
I love pasta            The new movie is so great       Score: 0.2560
I love pasta            The new movie is awesome        Score: 0.2440
A man is playing guitar         The cat plays in the garden     Score: 0.2105
The new movie is awesome        Do you like pizza?      Score: 0.1969
The new movie is so great       Do you like pizza?      Score: 0.1692
The cat sits outside            A woman watches TV      Score: 0.1310
The cat plays in the garden     Do you like pizza?      Score: 0.0900
```

Fig 9.5 Chunk Multiple Examples of Vector Space Model

Information Retrieval Systems
Vector Space Model in IR

- Such method is faster than brute force methods due to fewer samples.
- In practical industrial scenarios, more attention is paid to the speed of pre-trained models, encoding methods, and data retrieval.
- For example, two-tower model (Yang et al., 2020) and Wide&Deep model (Cheng et al. 2016) etc. are shown in Figs 9.6 and 9.7.

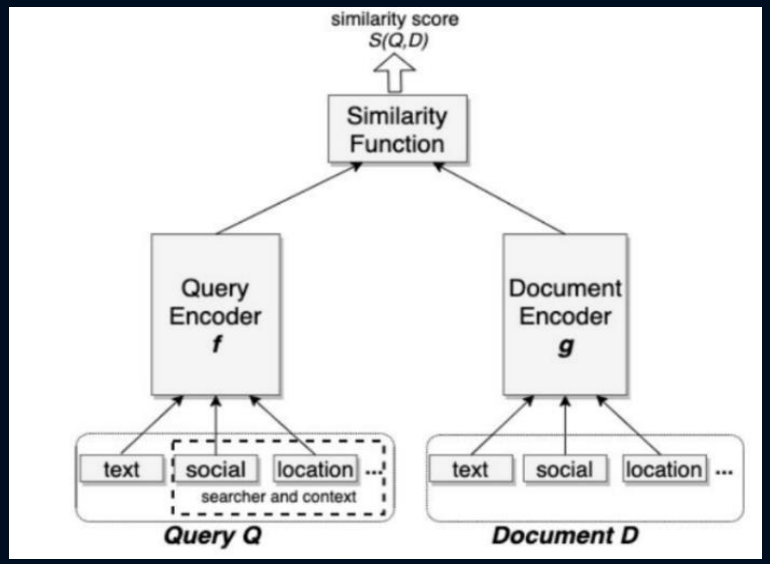


Fig 9.6 Two-Tower Model (Yang et al., 2020)

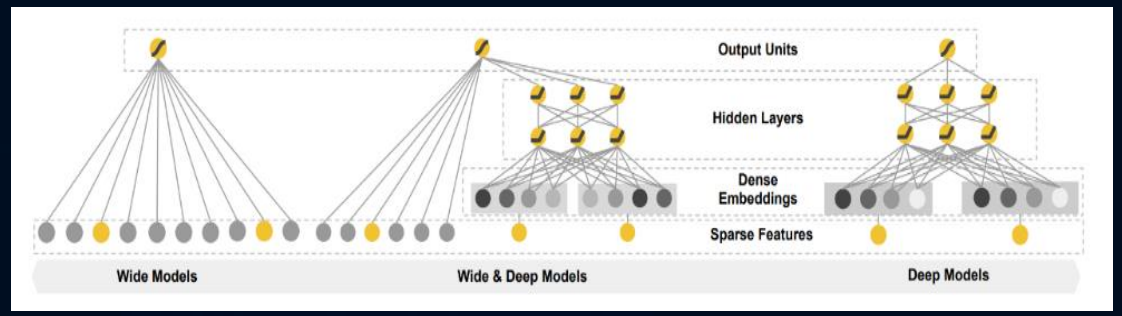


Fig 9.7 Wide&Deep model (Cheng et al., 2016)

Information Retrieval Systems
Term Distribution Models in IR

- Probabilistic Ranking Principle (PRP) models firstly proposed by Croft and Harper in 1979 (Croft & Harper, 1979) to compute query relevance degrees and retrieval. PRP regards IR as a process of statistical inference, where an IR system predicts query relevance from retrieved documents and sorts in descending order based on predicted relevance scores.
- This approach is like Bayesian model machine learning. A PRP model combines relevant feedback information with IDF and estimate each item's probabilities to optimize search engine retrieval performance.
- However, it is a difficult task to estimate each probability accurately in practical applications. Okapi BM25 (Whissell and Clarke, 2011) retrieval model had solved the difficulties encountered by PRP model with satisfactory performance in TREC retrieval experiments and commercial search engines.
- Many IR researchers had modifications based on BM25 model resulting in many variations, the most common form is as follows:

$$\text{sim}(Q,D)=\sum_{q\in Q}\log\frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)}\cdot\frac{(k_1+1)f_i}{K+f_i}\cdot\frac{(k_2+1)qf_i}{k_2+qf_i} \quad (9.2)$$

- There are two approaches to consider which is the best BM25 method:
  1) BM25 + Word2Vec embedding across all documents.
  2) BM25 + BERT + Word2Vec embedding for each top-k documents, select the most similar sentence embedding across top-k paragraphs.

Information Retrieval Systems
Term Distribution Models in IR

- Transformer-based neural network models are popular NLP research areas on enhanced parallelized processing capabilities. BERT is amongst one that uses Transformer-based deep bidirectional encoders to learn contextual semantic relationships between lexical items and performed satisfactory in many NLP tasks.
- It began to retrieve document with the most relevant documents followed by paragraphs and extract sentences from selected paragraphs. BERT embeddings are used to compare query with paragraphs and select the one with higher cosine similarity. Once relevant paragraphs are available, select sentence with answer by comparing sentence embeddings based on Word2Vec embeddings trained on the whole dataset, then average word embeddings in the paragraph with BM25 score calculation is shown in Fig. 9.8.

```
from get_result import filtered_query, remove_punct
from ranking import Ranking

# We can calculate and plot the scores of the documents talking about COVID19
ranking = Ranking()
covid_documents = df[(df.after_dec == True) & (df.tag_disease_covid == True)].paper_id

query1 = filtered_query(query1)
scores1 = ranking.get_bm25_scores(query1, covid_documents)

query2 = filtered_query(query2)
scores2 = ranking.get_bm25_scores(query2, covid_documents)

# Plot the results
fig, axs = plt.subplots(1,2, sharey=True, tight_layout=False, figsize =(15,5))
axs[0].hist(scores1.values(), bins=20, color='g')
axs[0].set_xlabel('Scores')
axs[0].set_ylabel('Number of documents')
axs[0].set_title('Incubation period')
axs[1].hist(scores2.values(), bins=20, color='g')
axs[1].set_xlabel('Scores')
axs[1].set_ylabel('Number of documents')
axs[1].set_title('Prevalence of asymptomatic shedding and transmission')

plt.show()
```

```
[nltk_data] Downloading package stopwords to /usr/share/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Fig 9.8 Sample code for Word2vec embeddings with BM25 score calculation

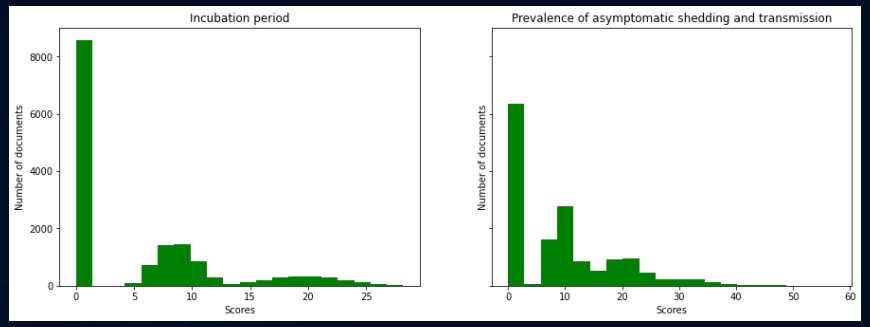


Fig 9.9 Documents Distribution with Scores and Codes

Information Retrieval Systems
Term Distribution Models in IR

- Common words queries occurred rarely in documents with a higher number of occurrences produce sparse distribution. Contrarily, there will be similar scores at many documents if common words with same frequency occurred across documents. Documents distribution with scores and codes are shown in Figs 9.9 and 9.10.
- Since word2vec relies heavily on each occurrence frequency, thus, it may pro-duce satisfactory performance on specific queries while the same for BERT on general queries.
- The results of two selected queries showed that query (Sentence 1) achieved satisfactory performance on specific/rare terminology while second query (Sentence 2) achieved satisfactory performance on normal terminology. They depend on words specification level in the query. For queries have specific/rare terminology performed satisfactorily with the most similar sentences across all documents. For queries have general terms e.g. age, human, climate performed satisfactorily with the most relevant documents instead of embeddings comparison across all of them. Thus, it is reasonable to compare each time the results of two approaches and select the appropriate one based on words distribution for each query.

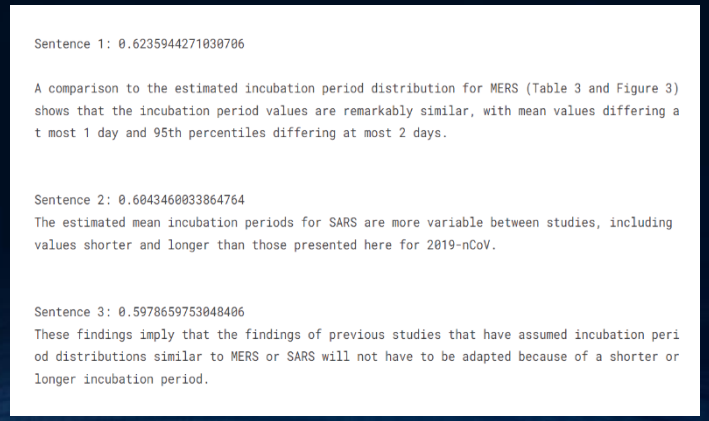


Fig 9.10 BM25 results

Information Retrieval Systems
Latent Semantic Indexing in IR

- Term Distribution Models in IR is a rapid and effective model. It uses topics to express implicit semantics of a document as index to replace incomplete, unreliable search terms by reliable indicants based on two assumptions:
  - Words have common topics in document.
  - Words not in document less likely to be related.
- Topic is filtered out by keywords in the Doc. Thus, P=(ω/Doc) probability distribution table is introduced: the statistics of word frequency (frequency) in the document i.e. the law of large numbers.

$$P(w \mid \text{Topic}_D) \approx P(w \mid D) = tf(w, D)/\text{len}(D) \qquad (9.3)$$

- *Topic* is regarded as a language model, and $P = (\omega/Doc)$ is the probability of word generation in this language model so the word not only occur in topic, but has probability generated.
- There are two methods of sorting according to statistical language model when query Q is given, which are 1) *Query-likelihood* and 2) *Document-likelihood* methods.

Information Retrieval Systems
Latent Semantic Indexing in IR

Query-likelihood

- Determine M_D, corresponding to each Doc, user's Query is denoted as Q = $(q_1, q_2, , , , q_n)$.
- Query probability will be generated under the *language model* of each document can be calculated as follows (Zhuang and Zuccon, 2021):

$$P(q_1 \dots q_k \mid M_D) = \prod_{i=1}^{k} P(q_i \mid M_D) \quad (9.4)$$

- Search results are obtained by sorting all computed results.
- However, this method calculates the probability for each Doc independently from other Docs, and the relevant documents are not utilized.

Information Retrieval Systems
Latent Semantic Indexing in IR

Document-likelihood

- Determine each Query corresponding M_Q. Calculate the probability that any given document will be generated under the query's *language model*. (Zhuang and Zuccon, 2021):

$$P(D \mid M_Q) = \prod_{w \in D} P(w \mid M_Q) \qquad (9.5)$$

- The object of one-mode factor analysis traditionally is a matrix composed of identical object-pair types of relationships.
- An example is a document-document matrix. The matrix elements may be evaluated for similarity between documents manually.
- This symmetric square matrix is decomposed into two matrices by eigen-analysis.
- The decomposed matrix is composed of linearly independent factors.
- Many of the factors are tiny that can be ignored usually producing an original matrix approximation.

Information Retrieval Systems
Discourse Segmentation in IR

- Document contents combine with articulated parts such as paragraphs exalt automatic documents segmentation according to meanings using machine learning methods to compare two adjacent sentences similarity in turn, and generate segmentation point with the lowest similarity.
- This unsupervised method is called Text Tiling (Hearst, 1997) as shown in Fig. 9.14.
- Further, supervised learning methods can also be used such as classifiers constructions (Florian, 2002) or sequence models (Keneshloo et al., 2019) to detect segmentation point.

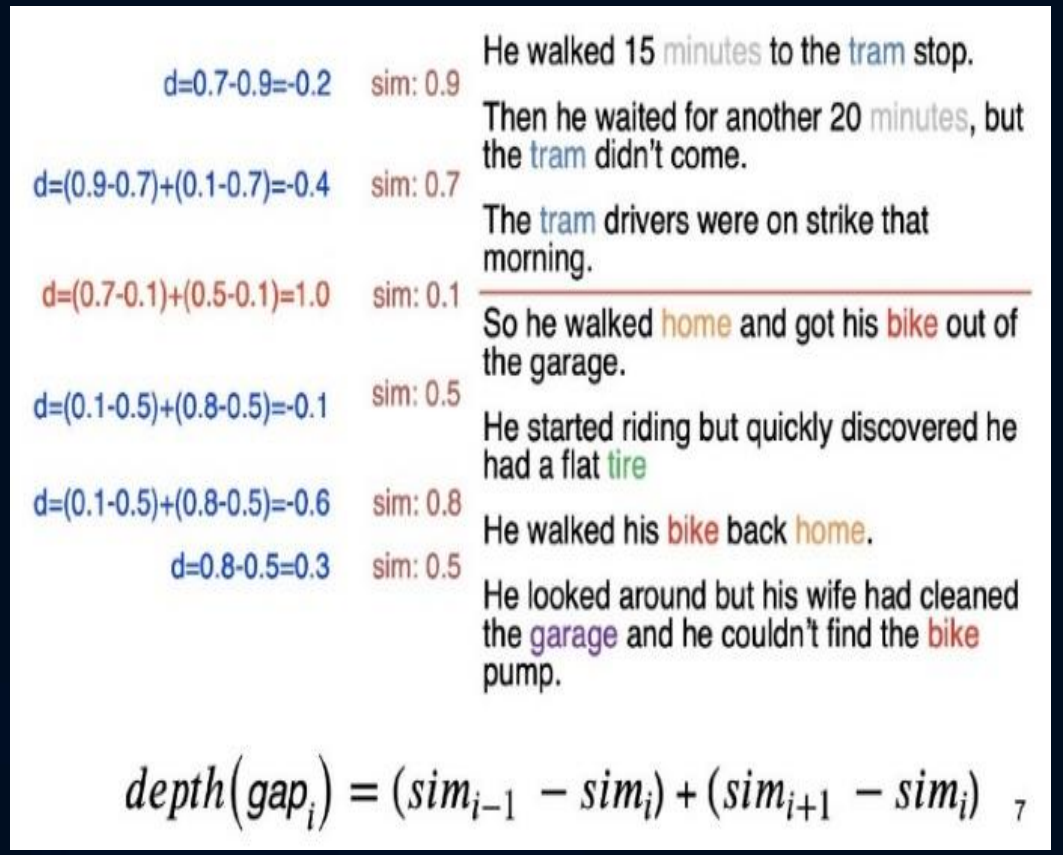


Fig 9.14 Examples of Discourse segmentation

Information Retrieval Systems
Discourse Segmentation in IR

- Rhetorical Structure Theory (RST) framework (Taboada, 2006) is a commonly used framework for parsing discourse as shown in Fig. 9.15. RST common relations in English are conjunction, justify, concession, elaboration, etc. as shown in Figs 9.16 and 9.17.
- Discourse segmentation task is a significant evaluation indicator for NLP development directions. From application perspective, discourse segmentation can assist users rely on intelligence to improve productivity, its technology core value can convert semi-structured and unstructured data to specific description structured in turn to support substantial downstream applications.

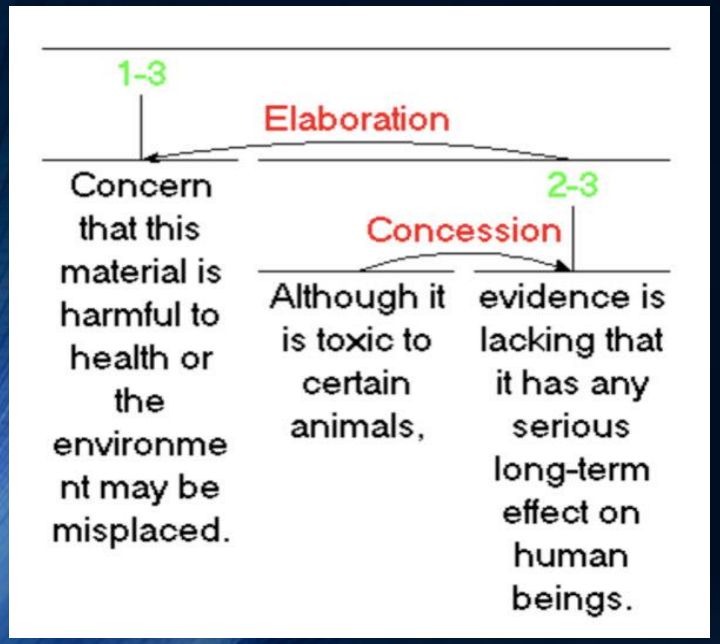


Fig 9.15 Example of Rhetorical Structure Theory

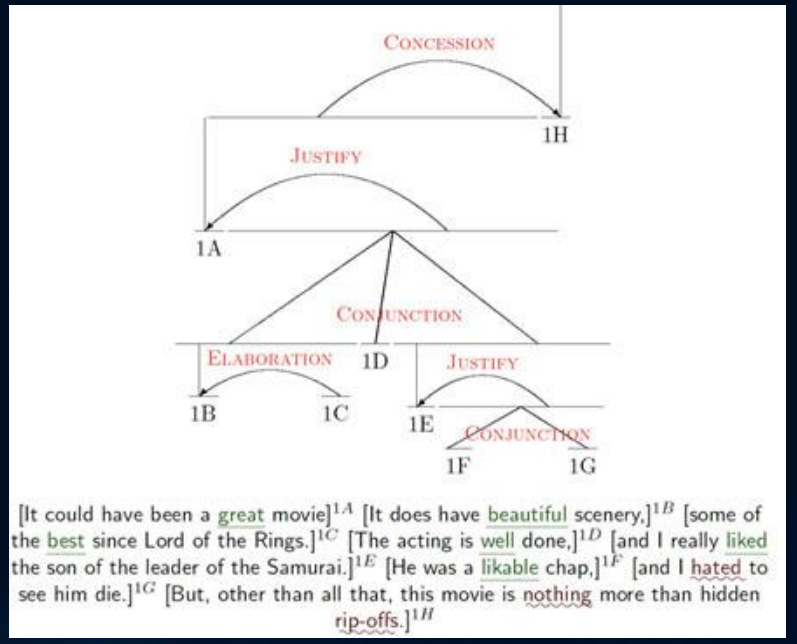


Fig 9.16 Examples of relations

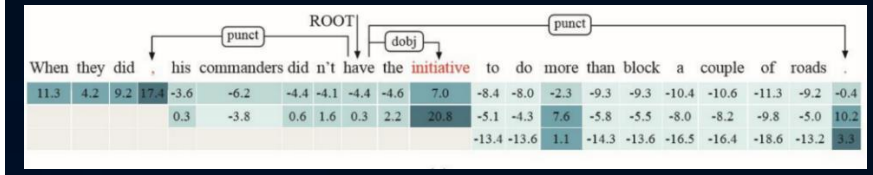


Fig 9.17 Attention map

Part III
Text Summarization Systems

Text Summarization Systems
An Introduction

Motivation

- There is excess information from copious sources to obtain the latest information daily.
- Although automatic and accurate summarization systems can assist users to simplify, identify and understand key information in the shortest possible time but they remain challenging as new words and complex text structure documents are available constantly.

Task Definition

- Text summarization process generates text (document or document) summaries by rewriting and summarizing long text into short form (Mahalakshmi and Fatima, 2022).
- It refers to extract or refine text or text set key points through technologies to display original text or text set main contents or general idea.
- Text generation task is an information compression technique whereas a summarization process is considered as a function where input is a document or documents, and output is an input texts summary.
- Hence, input and output are quintessential types to classify summary tasks.

Text Summarization Systems
An Introduction

Basic Approach

- Summarization approaches are mainly divided into *extractive* and *abstractive* (Chen et al 2018).
- **Extractive methods** select important phrases from input text, combine them to form a summary like a copy and paste process. Many traditional text summarization methods use **Extractive Text Summary (ETS)** because it is simple to generate sentences without grammatical errors but cannot reflect exact sentences meanings. They are inflexible to use novel expressions, words, or connectors outside text descriptions.
- **Abstractive Text Summary** (ATS) methods use language generation methods to re-organize contents, generate new words and conclude the implied information as compared with ETS. They paraphrase text meanings composed of new words with original words summary (Agrawal, 2020), and mimic human understanding to develop contents which may not contain in actual document text (Malki et al., 2020).

Text Summarization Systems
Task Goals in TS

Task Goals

- Summarization task objectives are to assist users to understand raw text within a short period as shown in Fig. 9.18.

Task Sub-processes

- Summarization tasks are divided into the following modules as shown in Fig. 9.19.
- Input document or documents are first combined and preprocessed from continuous text form to split sentences.
- The sentences will be encoded into vectors form data to fit into a matrix for similarity scores calculation to obtain sentence rankings, followed by a summary with the highest possibility according to the ranking list.

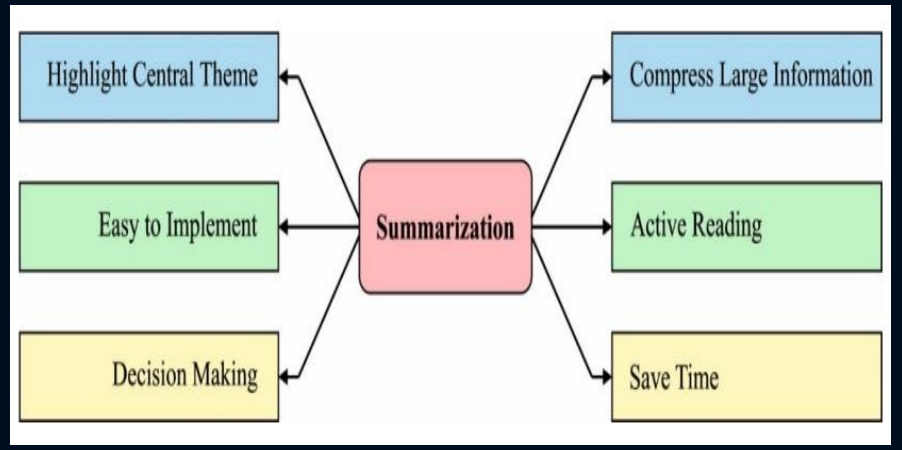


Fig 9.18 Summarization tasks objectives

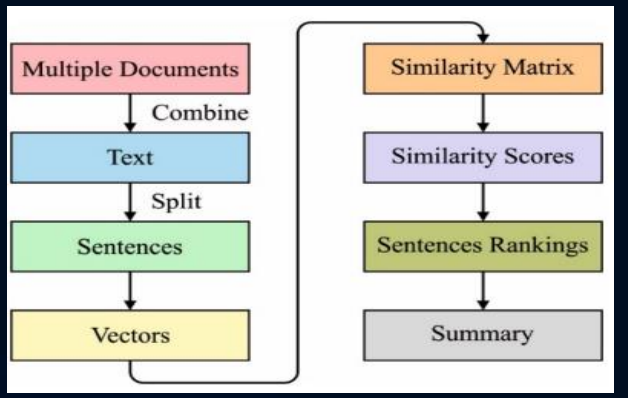


Fig 9.19 Summarization tasks sub-processes

Text Summarization Systems
Text Summarization Datasets

- Text summarization datasets commonly used include DUC (DUC,2022), New York Times (NYT, 2022), CNN/Daily Mail (CNN-DailyMail, 2022), Gigaword (Gigaword, 2022) and LCSTS datasets (LCSTS, 2022).
- DUC datasets (DUC, 2022) are the most fundamental text summarization datasets developed and used for testing purposes only. They consist of 500 news articles, each with four human-written summaries.
- NYT datasets (NYT, 2022) contain articles published in New York Times between 1996 and 2007 with abstracts compiled by experts. The abstract datasets are sometimes incomplete and sporadic short sentences with average of 40 words.
- CNN/Daily Mail datasets (CNN-DailyMail, 2022) are widely used multi-sentence summary datasets often trained by generative summary system. They have a) anonymized version to include entity names and b) non-anonymized version to replace entities with specific indexes.
- Gigaword datasets (Gigaword, 2022) are abstracts comprising of the first sentence and article title with heuristic rules of approximately 4-million articles.
- LCSTS datasets (LCSTS, 2022) are Chinese short texts abstract datasets constructed by Sina Weibo (Weibo, 2022).

Text Summarization Systems
Types of Summarization Systems

- Text summarization task for input documents can be divided into two types:
  1. Single document summarization considers each input is one document.
  2. Multiple document summarization considers input has several documents.
- Text summarization task viewpoint can be divided into three classes:
  1. Query-focused summarization adds viewpoint to query.
  2. Generic summarization is generic.
  3. Update summarization is a special type which sets *difference* (update) viewpoint.
- Summarization systems based on contents can be divided into four types:
  1. Indicative Summarization describes contexts without revealing details especially the endings, it contains partial information only.
  2. Informative Summarization contains all information in a document or documents.
  3. Keyword Summarization reveals output generation is sporadic text which contains phrases or words of input documents.
  4. Headline Summarization is usually single line summary.
- These summarization systems can be divided according to summary languages such as Arabic (Elsaid et al., 2022), Chinese (Yang et al., 2012), English and Spanish summarization systems etc.

Text Summarization Systems
Query-focused vs Generic Summarization Systems

- Text summarization can be query-focused or generic.
- Summary associative with query shows document contents is relative to initial search query.
- A query-related summary generation is a process of retrieving query-related sentences/paragraphs from a document that has a strong similarity to text retrieval process.
- Hence, abstracts relevant searches are often undertaken by extending traditional IR techniques with many text abstracts in the literature fall into this category.
- A general summary, on the other hand, provides an overall sense of the document's contents.
- A proper general summary should cover main topics and minimize redundancy.
- Since there are no queries or topics to feed into summarization process, it is difficult to develop a high-quality general summarization method for evaluation (Gong and Liu, 2001).

Text Summarization Systems
Query-focused Summarization Systems

- Query-focused Summarization (QFS) is primarily addressed using extractive methods to produce text lacks coherence. QFS applied abstractive methods can overcome these limitations and improve incoherent texts availability.
- A Relevance Sensitive Abstractive QFS (RSA-QFS) framework (Baumel et al., 2018) is shown in Fig. 9.20.
- This model assumes that a trained abstractive model includes reusable language knowledge to accomplish QFS tasks. Methods of enhancing this pre-trained single document abstraction model with explicit modelling of query dependencies are studied to improve multiple input documents operating ability and adjust generated abstractions lengths accordingly

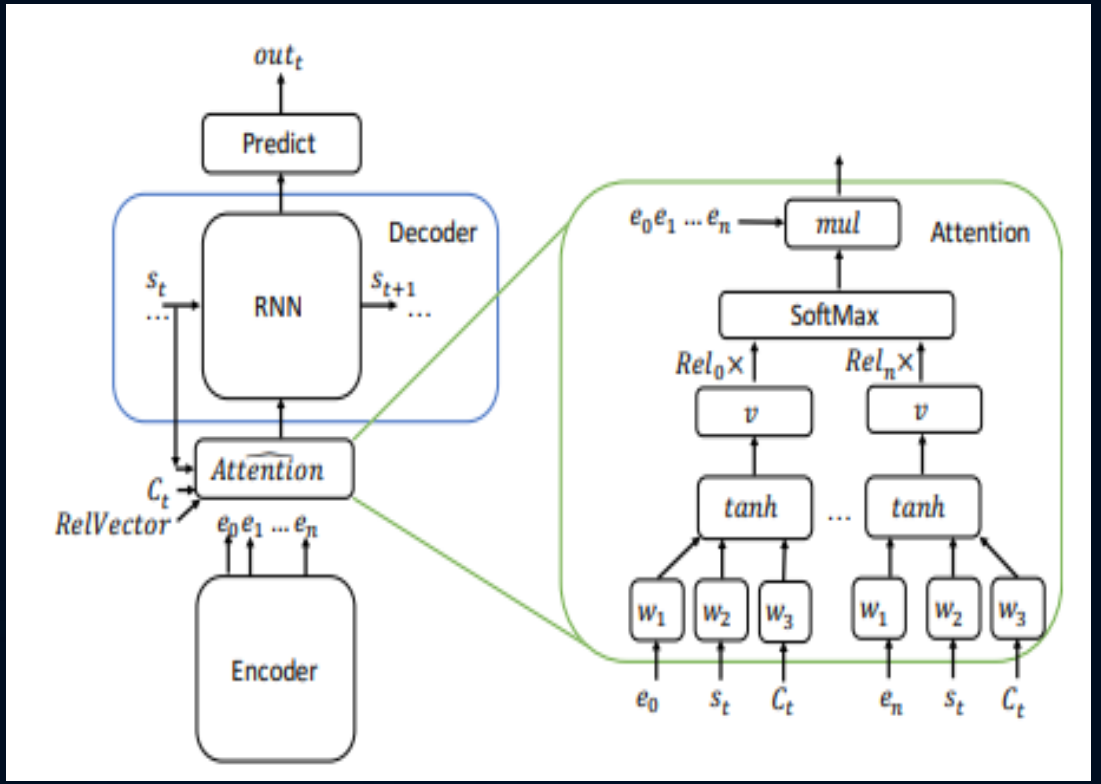


Fig 9.20 RSA-QFS framework

Text Summarization Systems
Query-focused Summarization Systems

- Further, a sequence-to-sequence (seq2seq) architecture is applied to obtain sum via an iterative extraction or abstraction pairs process: identify relevant content batches from multiple documents and abstract into a coherent text segments sequence.
- QFS task includes two stages as shown in Fig. 9.21:
  1. a relevance model to determine passages relevance to input query from source documents and
  2. a generic summarization method to combine relevant passages into a coherent summary
- Query-related text summarization are practical for answering questions such as whether a whole or partial document has relevance to a user's query.
- Query-related summaries do not provide an overall sense of the document's content, they have query bias and unsuitable for content summaries to answer questions such as document category, key points, text summary etc.

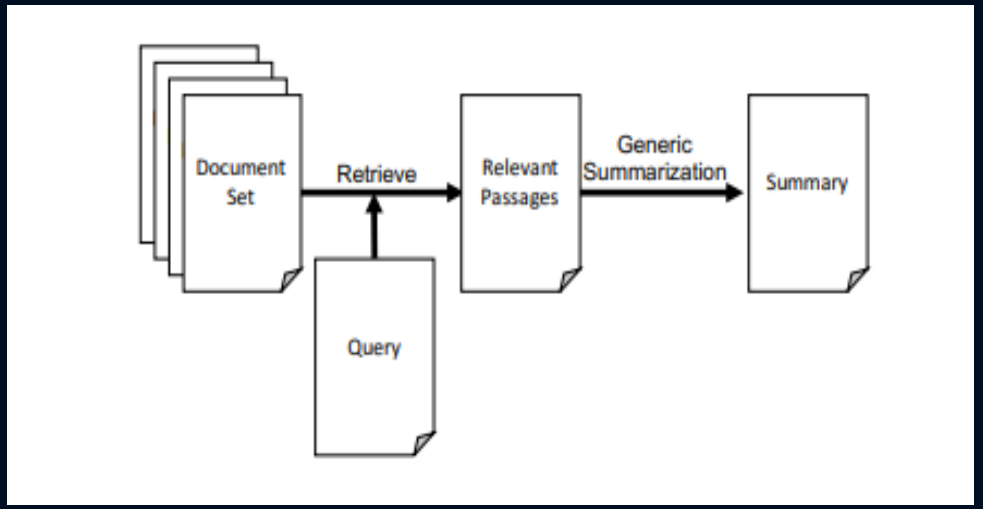


Fig 9.21 Two stages of QFS

Text Summarization Systems
Generic Summarization systems

- A proper generic summarization should cover main topics as many as possible and minimize redundancy leading to fractious system generation and evaluation.
- It often lacks consensus on summary output and performance judgments without query provisions and topics to summary task.
- Typical generic summarization ranking models and selected sentences are based on relevance similarity values and other semantic analysis (Gong and Liu, 2001).

Text Summarization Systems
Single and Multiple Document Summarization

- Single document extraction in journalism has developed to multi-document extraction since 1990. A variety of news articles, such as Google News (Google, 2022), Columbia News Blaster (Columbia, 2022) and News Essence (NewsInEssence, 2022) are inspired by multi-document summaries. The reason is that individual documents always produce contradictory results through overlapping information from multiple documents (Alami et al., 2015) may affect the performance of summarization results. Single document summarization research method gradually faded in past decades (Svore et al., 2007) as mainstream research focused on multi-document summarization which could reduce text size, gather ideas, compare documents, maintain syntactic and semantic relationships (Pervin and Haque, 2013).
- *Multiple document summarization* similarity measures and extractive techniques are comparable to single document summarization. It used clustering to identify common themes (Erkan and Radev, 2004), composite sentences from clusters (Barzilay et al., 1999), maximal marginal relevance (MMR) (Carbonell and Goldstein, 1998) and concatenated to multilingual environment (Evans 2005). Further, TFI X IDFI techniques (Salton 1989), TF/IDF (Fukumoto 2004), word hierarchical technique for frequent terms (You et al., 2009), graph-based methods (Mani and Bloedorn 1997; Wan 2008), sentence co-relation method (Hariharan et al., 2013), logical closeness (Zhu and Zhao, 2012) and query-oriented approach (Agarwal et al., 2011) are well-developed.

Text Summarization Systems
Contemporary Text Summarization Systems

Contemporary Extractive Text Summarization (ETS) System

- Text summarization research methods aim to (Dong, 2018):
  1. acquire important sentences.
  2. predict sentence option according to ranking sentences.
- The extractive summarization for proper sentences selection from original source text are required to:
  1. include logical and consistent summary information from original text.
  2. reduce similar and unimportant sentences information redundancy.
- Lead 3 is a commonly used and effective method to extract the first three sentences as topic titles of an article. When dealing with important sentences, document equivalence to document topic and relevant sentences position characteristics are considered. Topic modelling, frequency-based models LSA and Bayesian are methods applied (Farsi et al., 2021).
- Extractive summarization produces incoherent summaries compared with manual ones, its shortcomings include unresolved anaphora, unreadable sentence order, lacks textual cohesion to extract salient information from long sentences. When the system focuses on a sentence, it extracts the entire sentence (Nallapati et al., 2017).

Text Summarization Systems
Graph-based Method TS Systems

Graph-Based Method

- Graph-based ranking algorithms are successfully used in citation analysis, link social networks' structure analysis and the World Wide Web.
- They generate graphs from input document and summary by considering the relationships between nodes (units of text) (Chi & Hu, 2021).
- TextRank (Mihalcea and Tarau, 2004) is a typical graph-based approach that has developed many models.
- A summarization of TextRank system to extract keywords from a sample text and graph are shown in Figs 9.22 and 9.23.

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.

Fig 9.22 Sample text

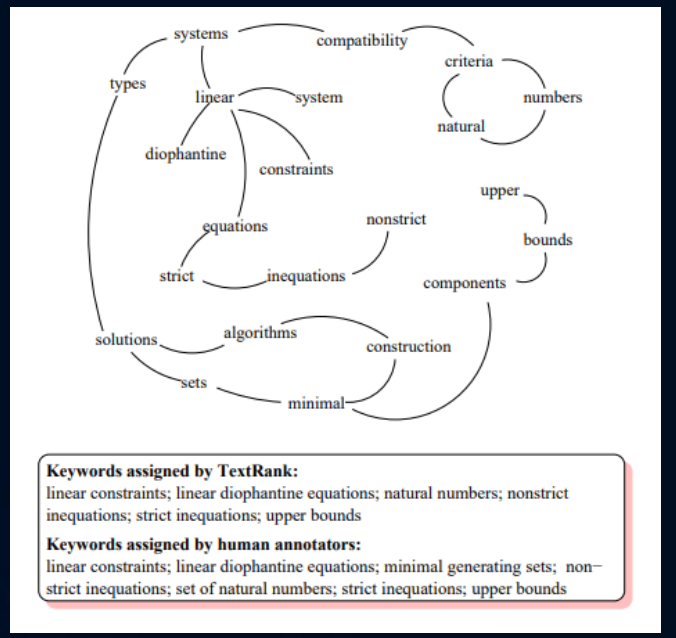


Fig 9.23 Sample graph for key phrase extraction in TextRank

Text Summarization Systems
Feature-based Method TS Systems

- Feature-based model extracts sentences features and evaluates their significances. There are many representative studies include Luhn's Algorithm (Luhn, 1958), TextTeaser and SummaRuNNer (Nallapati et al., 2017).
- Luhn's Algorithm is used to evaluate input words significance calculated by frequency. TextTeaser is an automatic feature-based summarization algorithm. SummaRuNNer is implemented by Deep Neural Networks (DNN) structure as shown in Fig. 9.25.
- SummaRuNNer generates sentence feature (vector) by two layers bi-directional Gate Recurrent Unit - Recurrent Neural Network (GRU-RNN) from word embed-ding vectors. The lowest level classifies each sentence word level, while the highest level classifies sentence level. Double arrows indicate two-way RNN. The top layer numbered with 1s and 0s is a classification layer based on sigmoid activation to determine whether each sentence is a summary. Each sentence decision depends on substantial sentence contents, sentences to document relevance, sentences to cumulative summary representation originality, and other positional characteristics.

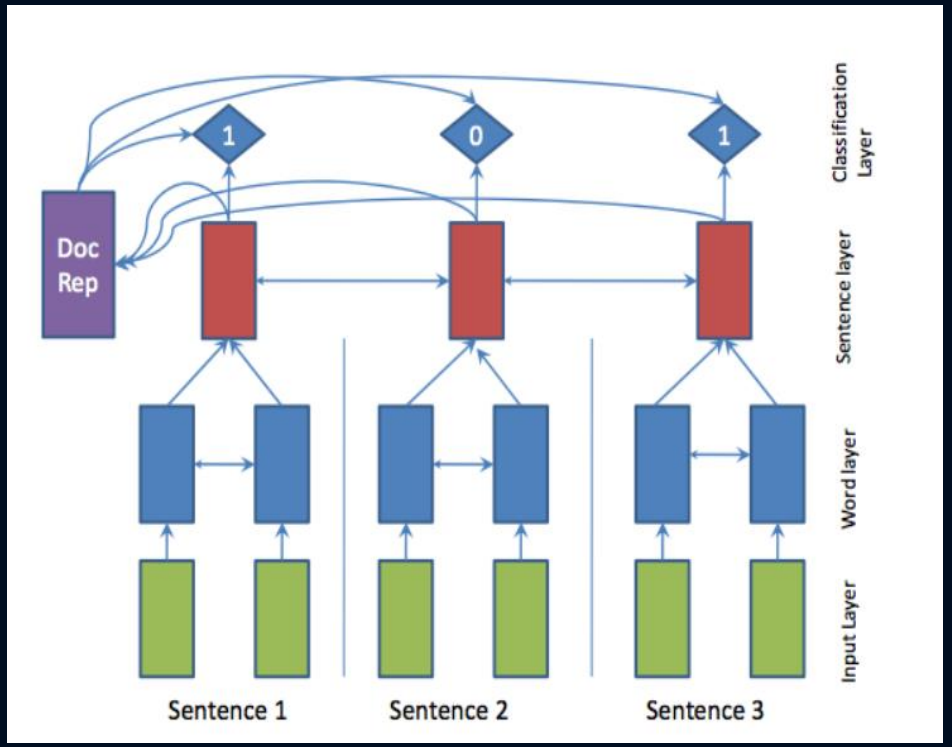
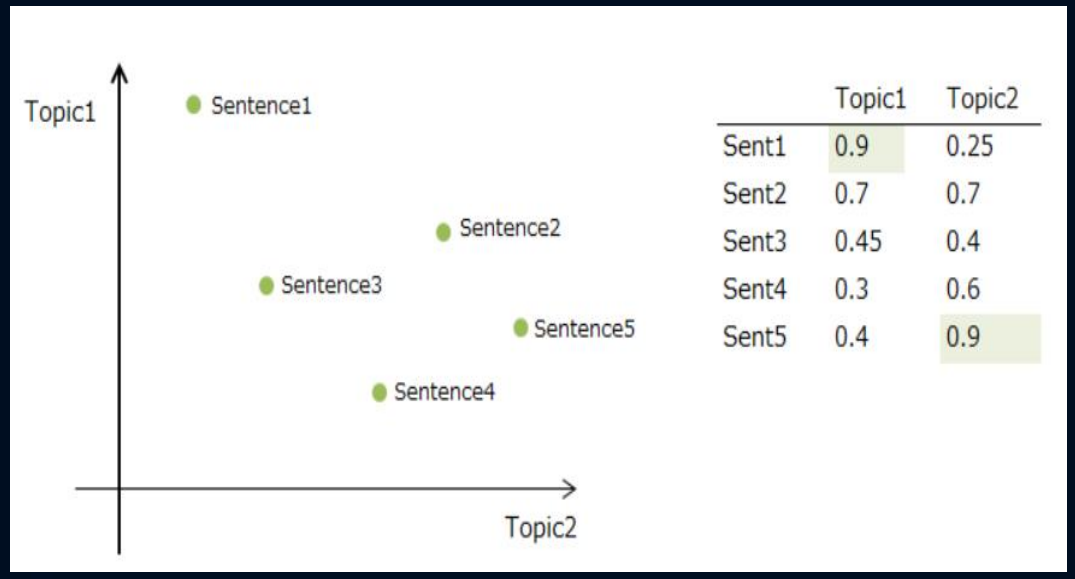


Fig 9.25 Network structure of SummaRuNNer

Text Summarization Systems
Topic-based Method TS Systems

- Topic-based model considers document's topic features and input sentences scores according to topic types contained as major topic would obtain a high rate when scoring sentences.
- Latent Semantic Analysis (LSA) is based on Singular Value Decomposition (SVD) to detect topics (Ozsoy et al., 2011).
- An LSA based sentence selection process is shown in Fig. 9.26 by topics represented by eigenvectors or principal axes with corresponding scores.



	Topic1	Topic2
Sent1	0.9	0.25
Sent2	0.7	0.7
Sent3	0.45	0.4
Sent4	0.3	0.6
Sent5	0.4	0.9

Fig 9.26 LSA based sentence selection sample

Text Summarization Systems
Grammar-based Method TS Systems

- Grammar-based model parses text and constructs a syntax structure, selects, or reorders the substructure.
- A representation framework is shown in Fig. 9.27.
- Grammar pattern can produce significant paraphrases based on grammatical structures.
- The above example in Fig. 9.28 showed how paraphrase extraction and replacement can be achieved by using such method.
- Analyzing grammatical structure feature is useful for semantic phrases reconstruction.

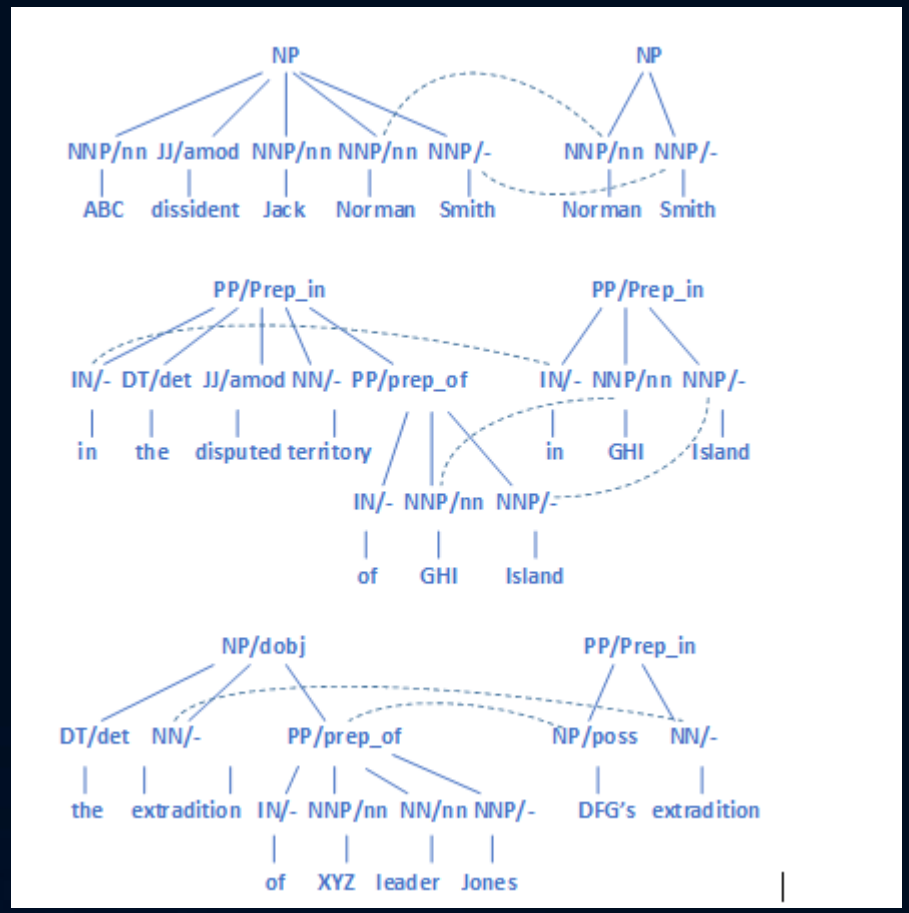


Fig 9.27 Grammar-based method sample network

Text Summarization Systems
Contemporary Abstractive Text Summarization (ATS) System

- Abstractive summarization often generates summary that maintains original in-tent completed by humans.
- This process can generate words that are not in original input representations but to facilitate summaries characteristics and fluency. However, it is complex to generate coherent phrases and connectors.
- Abstractive summarization systems applying deep learning methods, Reinforcement Learning (RL), Transfer Learning (TL) and Pre-Trained Language Models (PTLMs) had developed rapidly (Alomari et al., 2022) in recent years. These models use rules-based frameworks to consider significant events and summaries. Tree methods are ontology-related methods for abstractions (Jain et al., 2020).

Aided Summarization Method

- This method combines automatic computer model or algorithm to provide significant document information for human decision.
- Machine translation model to text summarization was proposed (Banko et al., 2000) applying encoder-decoder framework as neural network model mainstream and used in abstractive summarization systems (Chopra et al., 2016).

Text Summarization Systems
Contemporary Combined Text Summarization System

- Pointer-Generator Networks (See et al., 2017) is a frequently used baseline net-work. It focuses on keywords and sentences with Attention technique (Vaswani et al., 2017), to lever generator and pointer network according to calculated probability. Vocabulary and attention with different weights distribution are then combined. A baseline pointer-generator network framework is depicted in Fig. 9.28.
- It noted that article tokens are fed into an encoder layer, which is a single layer bidirectional LSTM with encoder hidden states provided. Decoder consists of a single-layer unidirectional LSTM, processes word embedding of previous words on each step and output decoder state with attention distribution.

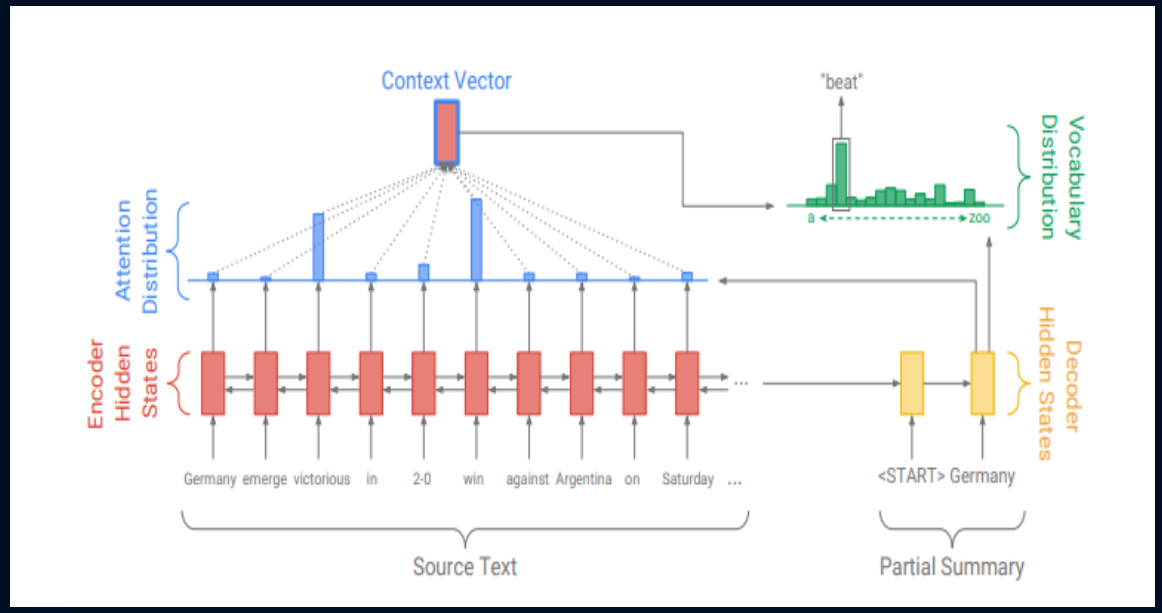


Fig 9.28 Network Framework of Point Generator Baseline Model

Summary

Summary

- This chapter study 3 vital NL applications: Information Retrieval Systems (IR), Text Summarization Systems (TS) and QA Systems.
- IR section summarized Vector Space Model baseline, Term Distribution Models, Latent Semantic Indexing, Discourse Segmentation, and sorts into multiple perspectives to identify the optimal solution in IR Systems.
- TS section is traditional NLP task to undergo developments supported by models, methods, and summarization task itself. Abstract generation is text generation with repetition, redundancy, incoherence, short generation with specific problems to determine key information. Text summarization today focuses on genuine summarization than sentences compression. Key words, external knowledge and other information are supplementary. Each abstract model has its own advantages as revealed in experimental results. Thus, it is crucial to address problems through evaluation index for practical model to meet with requirements.
- QA system section studied the basic concepts and requirements of a QA system in NLP with reviews on commonly used ones such as Microsoft QA system, AliMe and Xiao Ice QA systems followed by how the latest Transformer and BERT technologies are applied to these systems. In chapter 16 - Building Chatbot with TensorFlow and Transformer Technology will practice on how to use Python TensorFlow and Transformer technology to implement a live QA chatbot system

Hope you enjoy this course