TARAS SHEVCHENKO NATIONAL UNIVERSITY OF KYIV
Faculty of Computer Science and Cybernetics
Department of Mathematical Informatics

QUALIFICATION WORK
for obtaining a master's degree

in a training direction 122 Artificial Intelligence
on the topic:

NATURAL LANGUAGE DIALOG SYSTEMS

Made by 2nd year student
Oleksii Potopeiko _____
(signature)

Academic adviser:
Ph.D., Docent
Taras Panchenko _____
(signature)

I certify that in this work there are no borrowings from the works of other authors without the corresponding references
Student _____
(signature)

The work was considered and allowed to be defended at the session of the Department of Mathematical Informatics
«__» _____ 202_y.,
protocol №____
Acting Head of Department
V. Tereshenko _____
(signature)

Kyiv – 2021

Table of Contents

INTRODUCTION

The circulation and use of information in all areas of life highly depends on the availability of computer networks that each day enclose the planet in tighter meshes. Radio, television, telephone, increasingly denser transport systems, the extension of telematics, satellite technology, and the computer playing a central role - break up today's borders between humans and bring them together in a new world of communication. This world allows users to access an increasing number of different databases including images and sounds, text content and multiple information available at different sites all over the world.

Today it is not necessary any more to move to different locations in order to read a book, a document or to obtain any information. Navigating in multimedia hypertexts allows users to circulate in a world where information is accessible to everybody. Nowadays it is enough to talk to a machine in natural language in order to obtain train or airplane schedule information, to book a seat or to buy a ticket. Though, there is still much space for improvement in this direction.

If we analyze a dialog of a telephone caller requesting information from an operator, we realize that the operator has certain physical (auditory, articulatory, etc.) and cognitive (comprehension, reasoning, etc.) capabilities. He is therefore able to hear and to understand, to contextually interpret the utterances of the caller, to seek the requested information from a terminal, to manage the communication appropriately and, finally, to provide a relevant response to the caller.

A human-machine interface represents an implementation of models for these understanding and dialog processes. To date, it is still not possible to build machines enabling a dialog with anybody about any subject, but we come closer to it every year. State-of-the-art technology is limited to interfaces able to communicate with a person using spoken natural language in order to provide the requested information within a limited application domain that is defined and fed by data using structured databases. Industry tools still can not work by directly deriving insights from unstructured data.

We can observe that in the world there are many different applications of audio-based human-machine interaction in different industries. Though, as a rule, they are intricately connected to the business needs of companies where it is used. Education system and its processes seems to be weakly affected or not affected at all by a progress in this field depending on the country, context, subject and area of knowledge.

Education process is still tightly connected to the people who share information and teach certain subjects. There are thousands or millions of books written for almost any subject that is present in school or university program. Optimal approaches for teaching and learning of different subjects were found during practice of millions of experts all over the world. These facts bring us to the point that some school or university subjects probably can be taught by a machine with applying all the formed knowledge base, deep learning and end-to-end speech recognition.

This paper aims to discover possible applications of natural language dialog systems backed by deep learning techniques in the education and implement a prototype of audio-based bot to assist in education process. Languages learning and English language in particular will be used for the current research as a subject of learning (teaching for a machine).

1. NATURAL LANGUAGE DIALOG SYSTEMS

1.1 Language

Object of a particular scientific discipline, language is primarily a practical issue that fills each moment of our life, including dreams, elocution or writing. Language holds a social function which becomes obvious when it is used: normal communication (conversation, information, etc.), oratory (political, theoretical, scientific, etc. speeches) and literature (spoken language folklore, written literature, prose, poetry, song, theatre, etc.).

Furthermore, the language influences large areas of the human activity. And if, in the normal communication process, we use the language almost automatically without paying particular attention to its rules, orators and writers are constantly confronted with this process, and handle it with an implicit knowledge of its laws, that science certainly has not yet totally detected.

Historically famous Greek and Latin orators dazzled and subjugated the crowds. It is well known that not only content and ideas influenced the audience, but the technique used by the orators to transmit these ideas using the language. Since the era of Ancient Greece, history has strongly relied on eloquence and rhetoric: Socrates, Platon, etc., are the precursors.

Language is a human property that allows to express and to communicate opinions and thoughts using a system of vocal or graphic signs. The language itself is a system of vocal signs used by a community of individuals to express themselves and to communicate. The language represents a social aspect of the individual; it seems to obey the social laws which shall be recognized by all the members of the community. Common to everybody, the language becomes a spoken language, the carrier of a unique message. The term *discourse* reflects the role of language in communication.

Language may be materialized by a succession of articulated sounds, a network of written marks, or even by a set of gestures. In fact, language may be apprehended like a system of communication signs between individuals or different communities. It therefore constitutes a more general discipline, called *semiotics*. Several meaning

systems seem to be able to co-exist without necessarily having language as their basis. Therefore, gestures, visual signs, as well as images, photography, cinema, painting and music may be considered as languages since they transmit on the basis of a specific code a message between two individuals or communities.

Language seems to be a particularly complex system in which different problems interfere. Given their complexity and diversity, language studies draw upon philosophy, anthropology, psychology, psychoanalysis, sociology, and various linguistic disciplines.

1.2. Dialog and Computer

Dialog is ubiquitous in our society; at all times we use dialogs to communicate, ask for a service, negotiate, dispute, joke, or even to lie and to mislead.

Ever since humans have been motivated to create machines that imitate their movements, functions and acts. The development of this increasingly complex type of machines, called *automaton*, seems to be based on a magic, religious, scientific or entertaining motivation. Therefore, throughout the centuries various mechanical automata have emerged with those of Jacques Vaucanson being the most known. In this constant evolution towards the imitation of the human by more sophisticated machines, a major event marked our time, the appearance of the computer along with a new discipline called Artificial Intelligence (AI) causing much interest in a constantly growing community. The famous logician Turing preceded the AI by raising the question is a machine able to think? It may be answered with the famous Turing test. And it is not by simple coincidence that Turing suggests establishing a dialog between a machine or a human on the one hand, and a person willing to learn about his interlocutor on the other hand. Maintaining verbal exchanges proves intellectual capabilities that are usually attributed to humans. Turing has been perfectly aware of this fact.

Before investigating the field of human-machine communication, it seems appropriate to define certain fundamental terms. Interaction is the mutual influence of two people. Conversation represents a particular type of interaction, i.e., vocal interaction. Any nonverbal interaction does not relate to conversation. For now we

define the term *dialog* as an exchange of verbal statements, uttered between two humans or between the human and the machine.

For quite a long time, computer science has been limited to study the use of programming or database access languages enabling to communicate with computers. The emerging field of microelectronics and the development of the computer technology enabled novice and inexperienced users to directly access computers without the support of computer specialists. We have therefore been witnessing the evolution of data processing applications along with a significant change of the human-machine interaction paradigms. Compared to the so-called *traditional* communication modes, including the manipulation of icons and text menus on computer screens and keyboards, the sequences of questions and answers or commands in an automated call-center, a real finalized co-operative dialog seems to be an alternative. The importance of a human-machine spoken language dialog that is as close as possible to spoken natural language seems needful, in the same manner as it now seems crucial to make information systems accessible to everybody.

1.3. Human-Machine Spoken Language Dialog

The spoken language introduces certain specificities into the dialog. These will be analyzed in the following sections. Before presenting the different knowledge sources that are necessary to a spoken language dialog system, the rules for a flexible and spoken natural language dialog should be introduced, as well as the architecture of a human-machine spoken language dialog system.

1.3.1 Speech and Human-Machine Interaction

The efficiency of spoken language is surprising: a non-experienced person is able to enter approximately 20 words using the keyboard, to write 24 words, and to utter on average 150 words per minute. Undoubtfully, speech represents the most natural way of communication. It enables hands-free eyes-free interaction and, in addition, allows to engage a spoken dialog via the telephone. Speech also allows to establish conversation in certain operational situations where time is a crucial factor,

such as in the air traffic control domain. In other words, speech is the most popular way of communication in our everyday life.

Experiments carried out by Chapanis (1979) have clearly shown the advantages of spoken communication in the accomplishment of a task: speed and reliability of the task execution (compared to the use of the keyboard input and screen output), a more natural, easy and spontaneous communication mode, the possibility to interfere with other communication modes, enabling a multi-modal communication.

In addition to these advantages, the use of speech seems crucial in certain situations to compensate for other human communication channels, if these are either saturated (e.g., a communication between the pilot and the aircraft), or inoperational in case of an handicap, e.g., for blind persons. All these advantages justify research in the area of automatic speech recognition, as well as its integration in human-machine spoken language dialog.

1.3.2 Specifics of Spoken Language Dialog

There exist considerable differences between written and spoken language: the writer is able to think about the formulation of his sentence. He may modify it until complete satisfaction. Similarly, the reader may read a sentence again in case of incomprehension or doubt. In turn, speech production errors may be corrected, but they cannot be eliminated. They need to be corrected in real time, which introduces hesitation, repetition and self-correction phenomena.

Human-machine spoken language dialog differs from written dialog primarily due to the limitations of current speech recognition systems and the intrinsic structure of the spoken language dialog. The limitation of speech recognition systems may be explained by the non-deterministic character of the recognition process including difficulties to account for short and degraded messages (e.g, hesitations, interjections, etc.). This limitation introduces a disturbing parameter into the understanding of messages, and thus into the dialog flow.

The intrinsic characteristics of spoken dialog include the spontaneousness of utterances sometimes yielding a significant amount of redundant information, repetitions, self-corrections, hesitations, contradictions, and even tendencies to stop

the interlocutor. They also include the non-grammatical structure of human utterances which is not only related to the spontaneousness of the utterance but also to the spoken natural language itself. Finally, they include clarification and/or reformulation sub-dialogs that depend on the limitations of the speech recognizer or the quality of the speech synthesis.

After having reviewed the specificities of the human-machine spoken language dialog, we now develop the rules for a natural and flexible dialog.

1.3.3 Rules for a Smooth-Spoken Language Dialog

Compared to dialogs between humans, the human-machine communication constitutes a completely different interaction mode. The dialog turns are well respected, and interruptions practically do not exist. Nevertheless, in order to obtain flexible dialogs, it seems necessary to establish a certain number of rules. In the following, we successively examine speech recognition, the management of the communication channel, the linguistic constraints, flexibility issues, the problems due to speaker adaptation and the meta-reasoning.

Speech recognition capabilities. Due to their non-determinism, speech recognizers introduce certain errors which may disturb the understanding process and, consequently, the human-machine dialog. In order to obtain an acceptable interaction, it seems necessary to use a speaker-independent recognizer that accounts for the various spoken language dialog phenomena, i.e., the hesitations, interjections, etc.

Communication channel management. The human-machine spoken language dialog does not leave, by definition, any trace. Therefore, to be able to manage the dialog, it seems necessary to generate system messages. These include, for example, the standby messages asking the interlocutor for patience (e.g., please standby), the restart messages (e.g., I listen to you, please go on) and messages to maintain the dialog that are useful for the communication (e.g., do not cross, wait).

Understand, interpret and deal with linguistic phenomena. In order to achieve a flexible communication, utterances need to be processed depending on their context. The utterance *I need the schedules for trains leaving tomorrow to Kyiv*

causes different system reactions depending on whether it is an information request or a reply to a previous question. This implies that semantic utterance analysis requires a contextual interpretation prior to the extraction of its intrinsic content. Furthermore, it seems necessary to account for the various linguistic aspects, including the language coverage at the vocabulary level and the authorized syntactic forms, as well as the processing of linguistic phenomena including ellipses, anaphors and synonymies. Anaphors are references to a word or to a concept quoted in the course of the dialog. They may be expressed by a pronoun, a demonstrative adjective, etc. (e.g., *draw a square, color it in red*). Ellipses are incomplete sentences, that require context information to make sense (e.g., *draw a square and a rhombus*). Synonymy is defined as an equivalence of different words in terms of their meaning.

Flexibility. The interlocutor freely expresses himself. This results in two types of mechanisms for dialog control. The first one should allow formulations that do not correspond to the syntactic constraints of the language. The second mechanism aims at according an active role to the interlocutor in the dialog, whilst providing sufficient guidance. This implies that the machine needs to be able to identify the general dialog topic, on the basis of which it infers the goal and the eventual plan of the interlocutor.

Adaptation to the interlocutor. Depending on the application, the response of the machine needs to be more or less adapted to the interlocutor or user. This property seems crucial in intelligent computer-assisted educational systems ***(EIAO)***. It seems important for interfaces accessible to the general public (accounting for the age and socio-cultural level of the interlocutor) and still interesting to be considered in expert support systems, whereby distinction should be made between the adaptation at the level of user expertise, and the adaptation in terms of familiarity with the system.

However, it seems impossible to determine the exact degree a machine needs to adapt to the human and, to what extent the user should accept the constraints imposed by the machine. However, linguistic studies have shown that humans tend to modify their linguistic behavior, by using stereotypes and by limiting the size of his vocabulary, without necessarily being aware of the language level the vocabulary

manageable by the machine. It has also been found, that in spoken language, humans tend to adapt their rate/rhythm and elocutionary speed to the machine

Meta-reasoning. Although this topic remains a research area, a natural dialog interaction implies that up to a certain degree the computer is aware of its own knowledge and capabilities or, in other words, has a certain representation of itself. This enables the system to be aware of its limitations and, consequently, to appropriately react to those user questions it is unable to answer. Although there exist differences between written dialog and spoken language, their cognitive levels are closely related. Therefore, if we do not focus on the speech recognition and communication channel, the high-level rules can be considered as valid for both written and spoken language dialog

1.3.4 Functions of the Spoken Language Dialog

After having reviewed some general rules for a flexible and natural human-machine spoken language dialog we now discuss the functions a dialog system needs to assume.

Communication channel management. It includes the generation of system utterances to manage the dialog.

Understanding and contextual interpretation. Both functions correspond to the integration of the syntaxis-semantic representation (the literal meaning) of the current user utterance into the discursive dialog representation, by resolving ambiguities, anaphors and ellipses.

Generation and synthesis. Both consist in generating and synthesizing system utterances comparable to those recognizable and understandable by the system.

Inference mechanisms. The introduction of reasoning allows to go beyond the stage of simple question and answer systems and therefore to deal with more complex dialogs.

Predictions. They enable the support of low-level processes, such as speech recognition and understanding. Since predictions make use of high-level knowledge (semantics, pragmatics) they contribute to the improvement of the dialog quality.

Dialog management. On the basis of the recognized and semantically interpreted utterance, the corresponding actions need to be triggered (i.e., asking and answering questions, etc.). This control function of the dialog manager requires an exhaustive representation of the end-to-end system and consequently an access to the various knowledge sources that will be analyzed in the following section.

1.3.5 Knowledge for Human-Machine Spoken Language Dialogs

Modeling human-machine spoken language dialog requires multiple knowledge about the language, dialog, task, user and the system itself. In general, distinction is made between the static knowledge that does not vary throughout the dialog, and the constantly evolving dynamic knowledge.

The static knowledge includes the following models:

Language model. Due to the technology limitations, state-of-the-art systems generally make use of distinct language models both for recognition and understanding (due to semantic and pragmatic aspects). The first model is used to determine the correct sequence of words on the basis of the phoneme string, whereas the second model is used to analyze and to interpret the user utterances. It should be noted that the understanding process makes frequently use of language models. These models may also be a basis for the response generation so that the system has identical linguistic abilities for both analysis and generation. The language model for the understanding process includes lexical, syntactic and semantic components.

Task model. This model includes the application-related knowledge, such as the manipulated objects or concepts, their interrelations, the inference rules that enable generating new knowledge, and the description of the task execution. The task model enables the system to interpret an utterance (spoken, written or multimodal) in the dialog context in order to accomplish a given task.

Dialog model. It provides a general description of the different application-dependent situations. It enables the system to contextually interpret the user utterances and to predict the dialog flow as well as authorized deviations from the main topic of conversation.

User model. This model administrates the knowledge about the user at all levels of the understanding process: phonological variants, stereotyped formulations, the user's point of view of the machine, his expertise in the application domain, etc.

System model. Still at a rather preliminary state of research, this model describes the knowledge of the system about its proper communication capabilities and its competence limitations (speech recognition, mouse, numerical glove, etc.). This model is particularly important for multimodal dialog systems. The dynamic knowledge about the language includes the following sources:

- **Task context.** It completes the task model in the case of an evolutionary application. The context includes task knowledge which is likely to change during the dialog. It enables the system to raise certain ambiguities and to interpret the user utterances.
- **Dialog history.** The exchanges between the user and the system, as well as their structures, are stored in a history. This allows inconsistency and speech recognition error detection to resolve anaphors, to process ellipses and, finally, to predict the subsequent system messages. The dialog history is necessary a real dialog, able to understand more than simple questions and answers.
- **User model.** In addition to the static user model, a dynamic model that evolves throughout the dialog and depends on the user utterances, goals, plans, etc., may be established. This model allows to adapt the dialog to the user by adopting adequate strategies, by altering style and level of the generated system utterances to those of the user and by choosing possible explanations. User modeling becomes especially relevant in intelligent computer-assisted systems and depends on the degree of the user knowledge. In this particular case, the model keeps track of how this knowledge evolves over time.
- **System model.** This model may also be updated throughout the dialog. It is subject to change depending on the state of the connected peripheral devices (e.g., activity recognition) and according to the knowledge of the system about its own capabilities and limitations.

1.4 Spoken Language Dialog System

An overview of a spoken language dialog system is shown in Figure 1.1. It contains components for speech recognition, semantic analysis, dialog management, an interface to data access and a system response generation component.

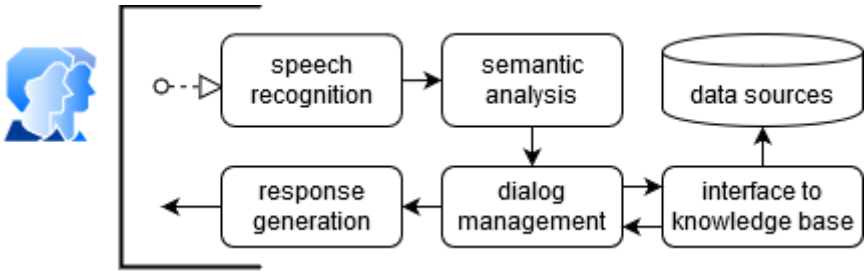


Figure 1.1 Spoken Language Dialog System diagram

The input utterance is recognized by a speech recognizer (an introduction into the problem of speech recognition is given by Rabiner (1989) and Young (1992)). The output is then provided to the semantic analysis, which determines the meaning of the utterance and builds an appropriate semantic representation. Human-machine interaction, such as information retrieval, is a matter of interactive problem solving. The solution is often built up incrementally, with both the user and the computer playing active roles in the conversation. Contextual understanding consists of interpreting the user query in the con- text of the ongoing dialog, taking into account common sense and task domain knowledge. Semantic representations corresponding to the current utterance are completed using the dialog history in order to take into account all the infor- mation given by the user earlier in the dialog. If this information is insufficient for database access, ambiguous or if the database does not contain the infor- mation requested, the dialog manager may query the user for clarification and feedback. A database access intertace uses the meaning representation to gen- erate a database query and to access the application back-end, i.e., a database. A system response generator presents the interaction result in the form of text, speech, tables or graphics.

2. DEEP LEARNING

It is crucial to make machine to understand what we say, in what context, what is the purpose of our words and what a respond should be. In order to make it possible scientific community took a look on ourselves and processes behind our consciousness, speech and way of thinking.

Deep learning is the concept being an answer to those tricky questions. It is also being on of the most discussed things in machine learning both among researchers and in the media. The idea of neural networks, and subsequently deep learning, gathers its inspiration from the biological representation of the human brain (or any brained creature for that matter).

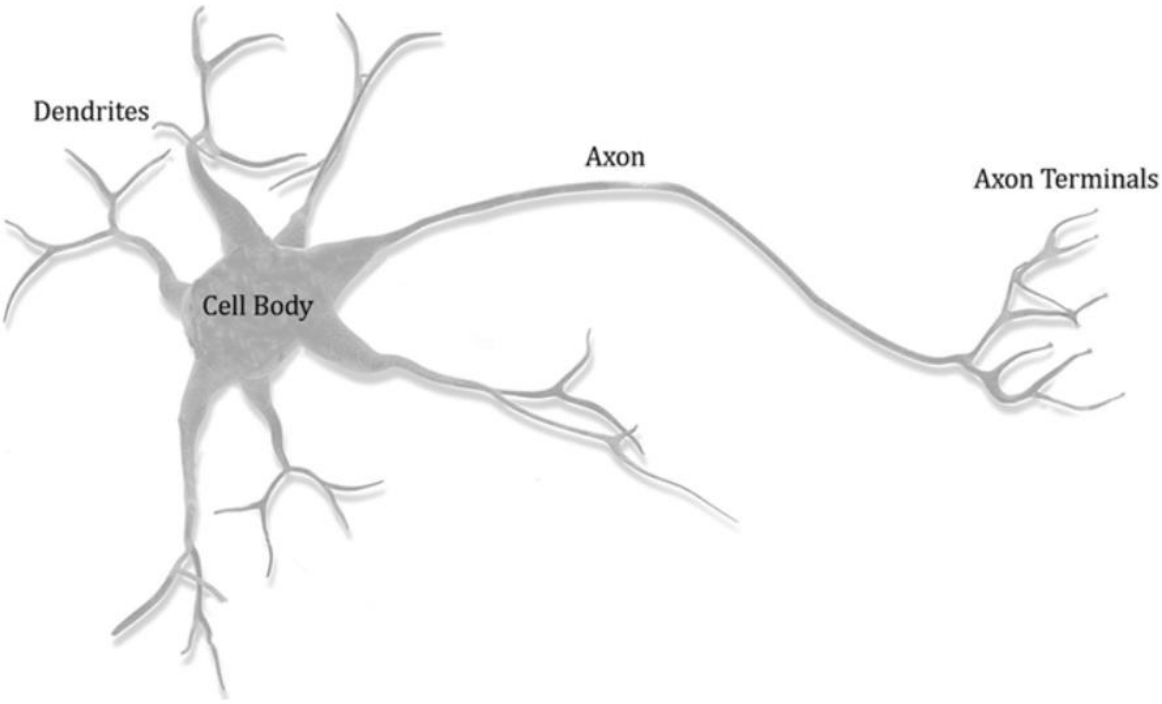


Figure 2.1 Diagram of biological neuron

The perceptron is loosely inspired by biological neurons (Fig. 2.1), connecting multiple inputs (signals to dendrites), combining and accumulating these inputs (as would take place in the cell body proper), and producing an output signal that resembles an axon.

Neural networks extend this analogy, combining a network of artificial neurons to create a neural network where information is passed between neurons (synapses),

as illustrated in Fig. 2.2. Each of these neurons learns a different function of its input, giving the network of neurons an extremely diverse representational power.

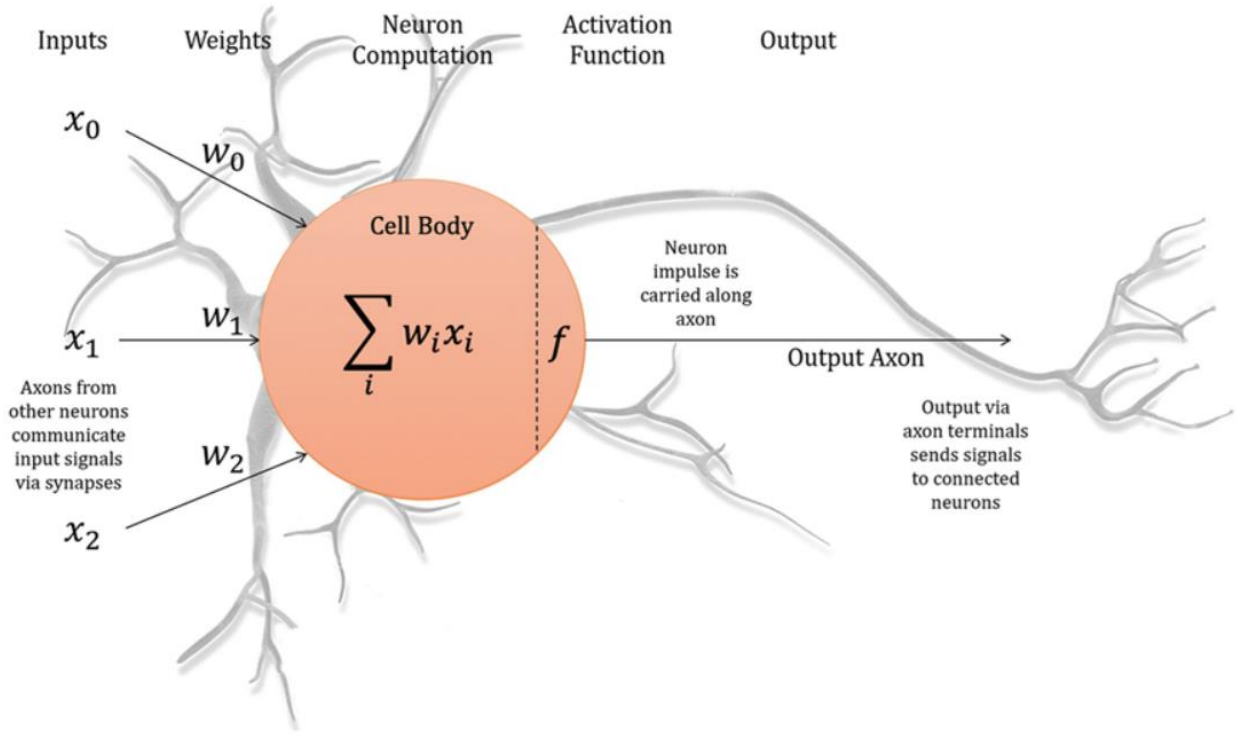


Figure 2.2 Diagram of an artificial neuron (perceptron)

The last 7–8 years have seen exponential growth in the popularity and application of deep learning. Although the foundations of neural networks can be traced back to the late 1960s, the AlexNet architecture ushered in an explosion of interest in the deep learning when it handily won the 2012 Imagenet image classification competition with a 5-layer convolutional neural network. Since then deep learning has been applied to a multitude of domains and has achieved state-of-the-art performance in most of these areas.

2.1 Distributed representations

While we aim to have a teacher that we can communicate to with an audio interface the foundation of any communication are words and there is a need for their representation in deep learning approaches. Here when it comes to the notion of *word embeddings* which serve to solve that problem of representation.

2.1.1 Distributional Semantics

Distributional semantics is a subfield of natural language processing predicated on the idea that word meaning is derived from its usage. *The distributional hypothesis* states that words used in similar contexts have similar meanings. That is, if two words often occur with the same set of words, then they are semantically similar in meaning. A broader notion is *the statistical semantic hypothesis*, which states that meaning can be derived from statistical patterns of word usage. Distributional semantics serve as the fundamental basis for many recent computational linguistic advances.

2.1.1.1 Vector Space Model

Vector space models (VSMs) represent a collection of documents as points in a hyperspace, or equivalently, as vectors in a vector space (Fig. 2.3). They are based on the key property that the proximity of points in the hyperspace is a measure of the semantic similarlity of the documents. In other words, documents with similar vector representations imply that they are semantically similar. VSMs have found

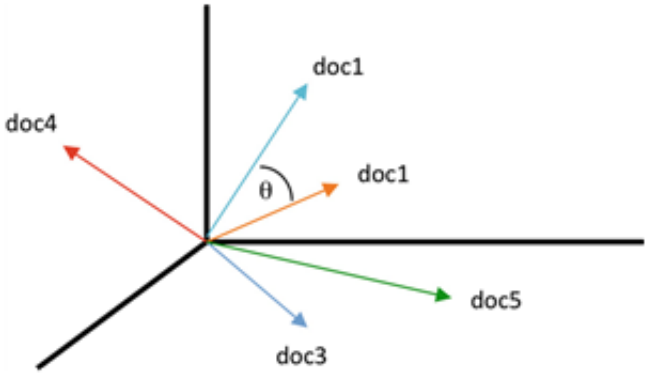


Figure 2.3 Vector space model representation for documents

widespread adoption in information retrieval applications, where a search query is achieved by returning a set of nearby documents sorted by distance.

2.1.1.1.1 Curse of Dimensionality

VSMs can suffer from a major drawback if they are based on high-dimensional sparse representations. Here, sparse means that a vector has many dimensions with zero values. This is termed the **curse of dimensionality**. As such, these VSMs require large memory resources and are computationally expensive to implement and use. For instance, a term-frequency based VSM would theoretically require as many dimensions as the number of words in the dictionary of the entire corpus of documents. In practice, it is common to set an upper bound on the number of words and hence, dimensionality of the VSM. Words that are not within the VSM are termed **out-of-vocabulary** (OOV). This is a meaningful gap with most VSMs in that they are unable to attribute semantic meaning to new words that they haven't seen before and are OOV.

The distributional hypothesis says that the meaning of a word is derived from the context in which it is used, and words with similar meaning are used in similar contexts.

2.1.1.2 Word representations

One of the earliest use of word representations dates back to 1986. Word vectors explicitly encode linguistic regularities and patterns. Distributional semantic models can be divided into two classes, co-occurrence based and predictive models. Co-occurrence based models must be trained over the entire corpus and capture global dependencies and context, while predictive models capture local dependencies within a (small) context window. The most well-known of these models, word2vec and GloVe, are known as word models since they model word dependencies across a corpus. Both learn high-quality, dense word representations from large amounts of unstructured text data. These word vectors are able to encode linguistic regularities and semantic patterns, which lead to some interesting algebraic properties.

2.1.1.2.1 Co-occurence

The distributional hypothesis tells us that co-occurrence of words can reveal much about their semantic proximity and meaning. Computational linguistics leverages this fact and uses the frequency of two words occurring alongside each other within a corpus to identify word relationships. **Pointwise Mutual Information** (PMI) is a commonly used information-theoretic measure of co-occurrence between two words *w1* and *w2*:

$$PMI(w_1.w_2) = log\frac{p(w_1, w_2)}{p(w_1)p(w_2)}$$

where *p(w)* is the probability of the word occurring, and *p(w1, w2)* is joint probability of the two words co-occurring. High values of PMI indicate collocation and coincidence (and therefore strong association) between the words. It is common to estimate the single and joint probabilities based on word frequency and co-occurrence within the corpus. PMI is a useful measure for word clustering and many other tasks.

2.1.1.2.2 LSA

Latent semantic analysis (LSA) is a technique that effectively leverages word co-occurrence to identify topics within a set of documents. Specifically, LSA

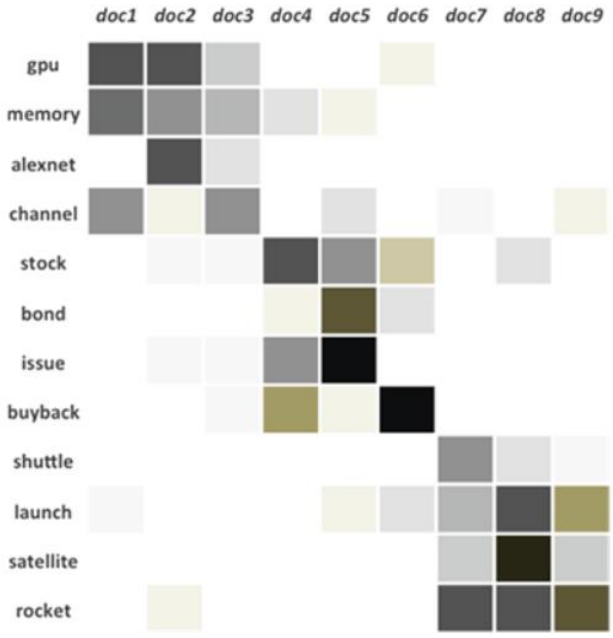


Figure 2.4 LSA document-term matrix

analyzes word associations within a set of documents by forming a document-term matrix (see Fig. 2.4), where each cell can be the frequency of occurrence or TFIDF of a term within a document. As this matrix can be very large (with as many rows as

words in the vocabulary of the corpus), a dimensionality reduction technique such as singular-value decomposition is applied to find a low-rank approximation. This low-rank space can be used to identify key terms and cluster documents or for information retrieval.

2.1.1.3 word2vec

In 2013, Tomas Mikolov proposed a set of neural architectures could compute continuous representations of words over large datasets. Unlike other neural network architectures for learning word vectors, these architectures were highly computationally efficient, able to handle even billion-word vocabularies, since they do not involve dense matrix multiplications. Furthermore, the high-quality representations learned by these models possessed useful translational properties that provided semantic and syntactic meaning. The proposed architectures consisted of the **continuous bag-of-words** (CBOW) model and the **skip-gram** model. They termed the group of models **word2vec**. They also proposed two methods to train the models based on a hierarchical softmax approach or a negative-sampling approach.

The translational properties of the vectors learned through word2vec models can provide highly useful linguistic and relational similarities. In particular, Tomas Mikolov revealed that vector arithmetic can yield high-quality word similarities and analogies. They showed that the vector representation of the word queen can be recovered from representations of king, man, and woman by searching for the nearest vector based on cosine distance to the vector sum:

$$v(\text{queen}) \approx v(\text{king}) - v(\text{man}) + v(\text{woman})$$

Vector operations could reveal both semantic relationships such as:

- $v(\text{Rome}) \approx v(\text{Paris}) - v(\text{France}) + v(\text{Italy})$

- v(niece) ≈ v(nephew) − v(brother) + v(sister)
- v(Cu) ≈ v(Zn) − v(zinc) + v(copper)
- as well as syntactic relationships such as:
- v(biggest) ≈ v(smallest) − v(small) + v(big)
- v(thinking) ≈ v(read) − v(reading) + v(think)
- v(mice) ≈ v(dollars) − v(dollar) + v(mouse)

Since wrod2vec models are state-of-the-art in the world today it's worth revisiting CBOW and skip-gram models. In the industry experts have found that CBOW models are better able to capture syntactic relationships, whereas skip-gram models excel at encoding semantic relationships between words.

2.1.1.3.1 CBOW

The **CBOW** architecture is based on a projection layer that is trained to predict a target word given a context window of *c* words to the left and right side of the target word (Fig. 2.5). The input layer maps each context word through an embedding matrix W to a dense vector representation of dimension *k*, and the resulting vectors of the context words are averaged across each dimension to yield a single vector of *k* dimension. The embedding matrix **W** is shared for all context words. Because word order of the context words is irrelevant in the summation, this model is analogous to a bag-of-words model, except that a continuous representation is used.

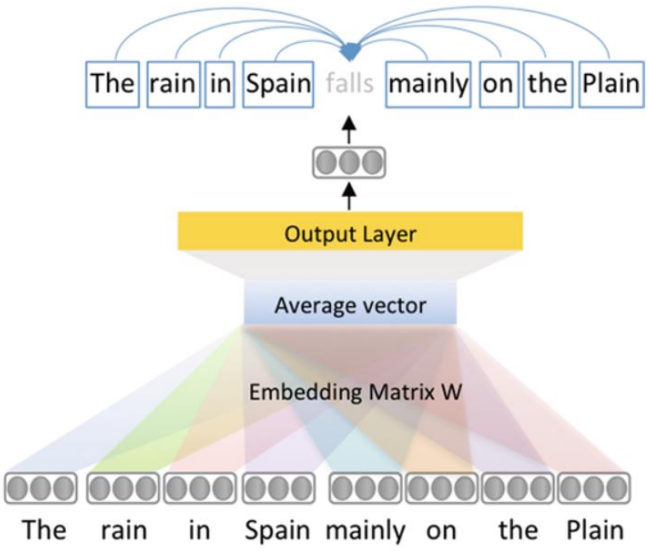


Figure 2.5 Continuous bag-of-words model (context window = 4)

The CBOW model objective seeks to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c<j<c,j\neq 0}\log\left(p\left(w_t\middle|w_{t+j}\right)\right)$$

where *c* is the number of context words to each side of the target word (Fig. 2.6).

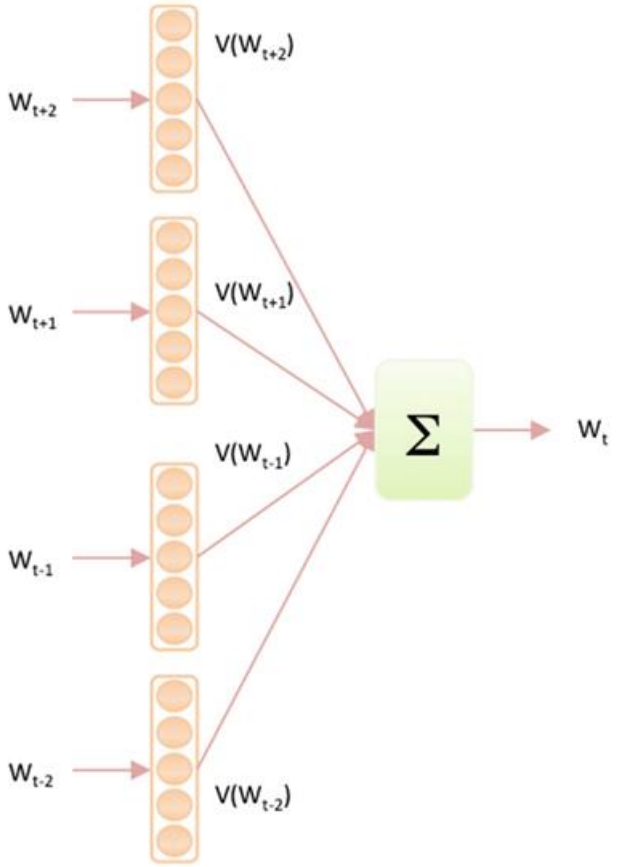


Figure 2.6 CBOW vector construction (context window = 2)

For the simple CBOW model, the average vector representation from the output of the projection layer is fed into a softmax that predicts over the entire vocabulary of the corpus, using backpropagation to maximize the log probability objective:

$$p\left(w_t\middle|w_{t+j}\right)=\frac{exp\ ({V'}_{w_t}{}^{T}V_{w_t+j})}{\sum_{w=1}^{V}exp\ ({V'}_{w}{}^{T}V_{w_t+j})}$$

where **V** is the number of words in the vocabulary. Note that after training, the matrix **W** are the learned word embeddings of the model.

2.1.1.3.2 Skip-Gram

Whereas the CBOW model is trained to predict a target word based on the nearby context words, the **skip-gram** model is trained to predict the nearby context words based on the target word (Fig. 2.7). Once again, word order is not considered. For a context size c, the skip-gram model is trained to predict the c words around the target word.

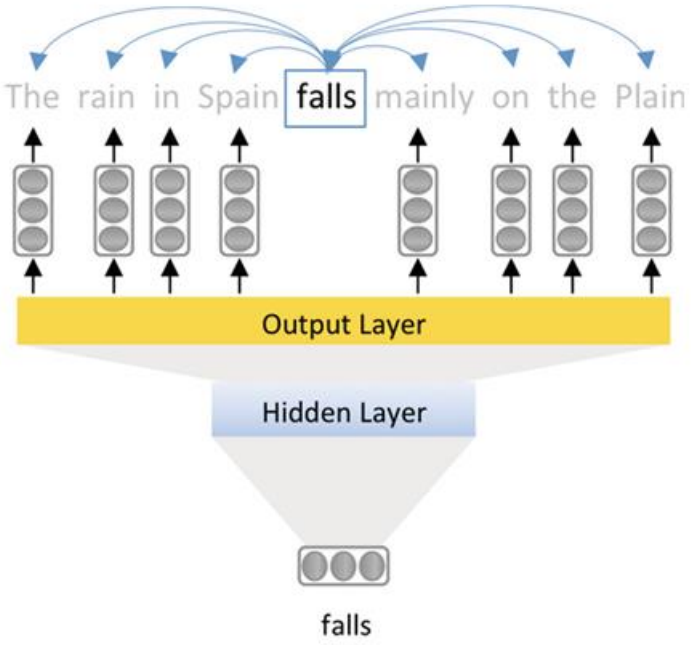


Figure 2.7 Skip-gram model (context window = 4)

The objective of the skip-gram model is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c<j<c, j\neq 0}\log\left(p\left(w_{t+j}\middle|w_t\right)\right)$$

where *c* is the size of the training context (Fig. 2.8). Higher values of c result in more training examples and thus can lead to a higher accuracy, at the expense of the training time. The most simple skip-gram formulation utilizes the softmax function:

$$p\left(w_{t+j}\middle|w_t\right) = \frac{exp\ ({V'}_{w_{t+j}}{}^{T} V_{w_t+t})}{\sum_{w=1}^{V} exp\ ({V'}_{w}{}^{T} V_{w_t})}$$

where *V* is the number of words in the vocabulary.

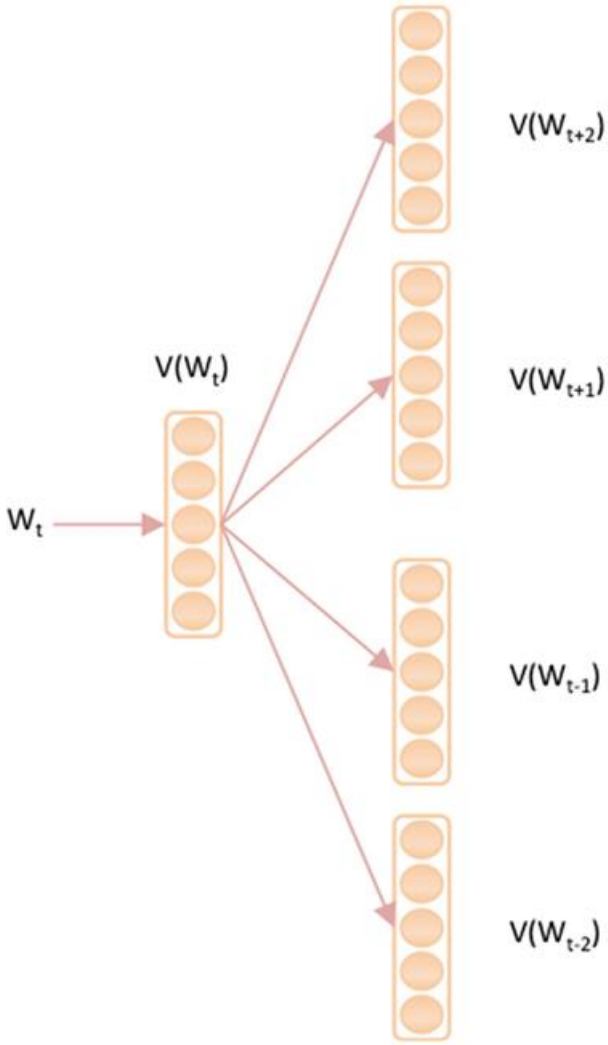


Figure 2.8 Skip-gram vector construction

It is interesting to note that shorter training contexts result in vectors that capture syntactic relationships well, while larger context windows better capture semantic relationships. The intuition behind this is that syntactic information is typically dependent on the immediate context and word order, whereas semantic information can be non-local and require larger window sizes.

2.1.1.4 GloVe

The global co-occurrence-based models can be the alternative to predictive, local-context window methods like word2vec. Co-occurrence methods are usually extremely high dimensional and require much storage. When dimensionality reduction methods are used like in LSA, the resulting representations typically

perform poorly in capturing semantic word regularities. Furthermore, frequent co-occurrence terms tend to dominate. Predictive methods like word2vec are local-context based and generally perform poorly in capturing the statistics of the corpus. In 2014, Pennington proposed a log-bilinear model that combines both global co-occurrence and shallow window methods. They termed this the **GloVe** model, which is play on the words Global and Vector. The GloVe model is trained via least squares using the cost function:

$$J = \sum_{i=1,j=1}^{V} f(X_{ij})(\boldsymbol{u}_i^T \boldsymbol{v}_j - \log(X_{ij}))^2$$

where V is the size of the vocabulary, X_{ij} is the count of times that words i and j co-occur in the corpus (Fig. 2.9), f is a weighting function that acts to reduce the impact of frequent counts, and $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ are word vectors.

	where	*in*	*the*	*sacred*	*river*	*ran*	*man*	*to*	*sunlit*	*sea*
where	0	2	1	0	0	0	1	2	0	0
in	2	0	0	1	0	0	1	0	1	0
the	1	0	0	4	3	1	5	0	2	1
sacred	0	1	4	0	2	0	1	0	0	1
river	0	0	3	2	0	3	0	1	0	0
ran	0	0	1	0	3	0	3	3	0	0
man	1	1	5	1	0	3	0	1	0	2
to	2	0	0	0	1	3	1	0	1	0
sunlit	0	1	2	0	0	0	0	1	0	2
sea	0	0	1	1	0	0	2	0	2	0

Figure 2.9 GloVe co-occurrence matrix (context windows = 3)

Typically, a clipped power-law form is assumed for weighting function f:

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{X_{max}}\right)^a & if\ X_{ij} < X_{max} \\ 1 & otherwise \end{cases}$$

with

$$X_{max}$$

is set at training time based on the corpus. Note that the model trains context vectors *U* and word vectors *V* separately, and GloVe embeddings are the given by the sum of these two vector representations $\mathbf{U} + \mathbf{V}$. Similar to word2vec, GloVe embeddings can express semantic and syntactic relationships through vector addition and subtraction. Furthermore, word embeddings generated by GloVe are superior to word2vec in performance over many NLP tasks, especially in situations where global context is important such as named entity recognition.

GloVe outperforms word2vec when the corpus is small or where insufficient data may be available to capture local context dependencies.

2.1.2 Limitation of Word Embeddings

Embedding models suffer from a number of well-known limitations. These include out-of-vocabulary words, antonymy, polysemy, and bias.

2.1.2.1 Out of Vocabulary

The Zipfian distributional nature of the English language is such that there exists a huge number of infrequent words. Learning representations for these rare words would require huge amounts of (possibly unavailable) data, as well as potentially excessive training time or memory resources. Due to practical considerations, a word embedding model will contain only a limited set of the words in the English language. Even a large vocabulary will still have many out-of-vocabulary (OOV) words. Unfortunately, many important domain-specific terms tend to occur infrequently and can contribute to the number of OOV words. This is especially true with domain-shifts. As a result, OOV words can have crucial role in the performance NLP tasks.

With models such as word2vec, the common approach is to use a “UNK” representation for words deemed too infrequent to include in the vocabulary. This maps many rare words to an identical vector (zero or random vectors) in the belief that their rarity implies they do not contribute significantly to semantic meaning. Thus, OOV words all provide an identical context during training. Similarly, OOV

words at test time are mapped to this representation. This assumption can break down for many reasons, and a number of methods have been proposed to address this shortfall.

Ideally, we would like to be able to somehow predict a vector representation that is semantically similar to either words that are outside our training corpus or that occurred too infrequently in our corpus. Character-based or subword (char-n-gram) embedding models are compositional approaches that attempt to derive a meeting from parts of a word (e.g., roots, suffixes). Subword approaches are especially useful for foreign languages that are rich in morphology such as Arabic or Icelandic. Byte-pair encoding is a character-based, bottom-up method that iteratively groups frequent character pairs and subsequently learning embeddings on the final groups. Other methods that leverage external knowledgebases (e.g., WordNet) have also been explored, including the copy mechanism that take into account word position and alignment, but tend to be less resilient to shifts in domain.

2.1.2.2 Antonymy

Another significant limitation is an offshoot of the fundamental principle of distributional similarity from which word models are derived—that words used in similar contexts are similar in meaning. Unfortunately, two words that are antonyms of each other often co-occur with the same sets of word contexts:

- I really hate spaghetti on Wednesdays.
- I really love spaghetti on Wednesdays.

While word embedding models can capture synonyms and semantic relationships, they fail notably to distinguish antonyms and overall polarity of words. In other words, without intervention, word embedding models cannot differentiate between synonyms and antonyms and it is common to find antonyms closely co-located within a vector-space model.

An adaptation to word2vec can be made to learn word embeddings that disambiguate polarity by incorporating thesauri information. Consider the skip-gram model that optimizes for an objective function:

$$J(\theta) = \sum_{w \epsilon V} \sum_{c \epsilon V} \{\#(w,c) log\sigma\big(sim(w,c)\big) + k\#(w) P_0(c) log\sigma(-sim(w,c))\}$$

where the first term are the co-occurrence pairs within a context window and the second term represents negative sampling. Given a set of synonyms

$$\mathbb{S}_\omega$$

and antonyms

$$\mathbb{A}_\omega$$

of a word *w*, we can modify the skip-gram model objective function to the form:

$$J(\theta) = \sum_{w \epsilon V} \sum_{s \epsilon \mathbb{S}_w} log\sigma\big(sim(w,s)\big) + \alpha \sum_{w \epsilon V} \sum_{s \epsilon \mathbb{A}_w} log\sigma\big(-sim(w,s)\big) + \sum_{w \epsilon V} \sum_{c \epsilon V} \{\#(w,c) log\sigma\big(sim(w,c)\big) k log\sigma(-sim(w,c))\}$$

This objective can be optimized to learn embeddings that can distinguish synonyms from antonyms. Studies have shown that embeddings learned in this manner to incorporate both distributional and thesauri information perform significantly better in tasks such as question-answering.

2.1.2.3 Polysemy

In the English language, words can sometimes have several meanings. This is known as **polysemy**. Sometimes these meanings can be very different or complete opposites of each other. Look up the meaning of the word *bad* and you might find up to 46 distinct meanings. As models such as word2vec or GloVe associate each word with a single vector representation, they are unable to deal with homonyms and polysemy. Word sense disambiguation is possible but requires more complex models.

In linguistics, word sense relates to the notion that, in the English language and many other languages, words can take on more than one meaning. **Polysemy** is the concept that a word can have multiple meanings. **Homonymy** is a related concept where two words are spelled the same but have different meanings. For instance, compare the usage of the word *play* in the sentences below:

- She enjoyed the play very much.
- She likes to play cards.
- She made a play for the promotion.

For NLP applications to differentiate between the meanings of a polysemous word, it would require separate representations to be learned for the same word, each associated with a particular meaning. This is not possible with word2vec or GloVe embedding models since they learn a single embedding for a word. Embedding models must be extended in order to properly handle word sense.

Humans do remarkably well in distinguishing the meaning of a word based on context. In the sentences above, it is relatively easy for us to distinguish the different meanings of the word *play* based on the part-of-speech or surrounding word context. This gives rise to multi-representation embedding models that can leverage surrounding context (cluster-weighted context embeddings) or part-of-speech (sense2vec).

2.1.2.3.1 Clustering-Weighted Context Embeddings

One approach to deal with word sense disambiguation is to start by building an inventory of senses for words within a corpus. Each instance of a word *w* is associated with a representation based on context words surrounding it. These representations, termed *context embeddings*, are then clustered together. The centroid of each cluster is the representation

$$S_{\omega_i}$$

for the different senses of the word:

$$sense(w_i) = arg \min_{j:\, s_j \in S_{w_i}} d(c_i, s_j)$$

where d is a distance metric (usually cosine distance). This can be implemented as the multi-sense skip-gram model (Fig. 2.10) where each word is associated with a vector **v** with context vectors **c** and each sense of the word is associated with a representation μ. Given a target word, a word sense is predicted based on $\boldsymbol{V}_{context}$:

$$s_t = arg \max_{k=1,2,\ldots K} sim\,(\mu(w_t, k),\ \boldsymbol{V}_{context}(\boldsymbol{c_t})$$

where *sim(a, b)* is a similarity function.

The multi-sense word embeddings are learned from a training set by maximizing the objective function:

$$J(\theta) = \sum_{(w_t, c_t) \epsilon D^+} \sum_{c \epsilon c_t} \log P(D = 1 | \boldsymbol{V}_s(w_t, s_t), \boldsymbol{V}_g(c)) + \sum_{(w_t, c\prime_t) \epsilon D^-} \sum_{c\prime \epsilon c\prime_t} \log P(D = 0 | \boldsymbol{V}_s(w_t, s_t), \boldsymbol{V}_g(c'))$$

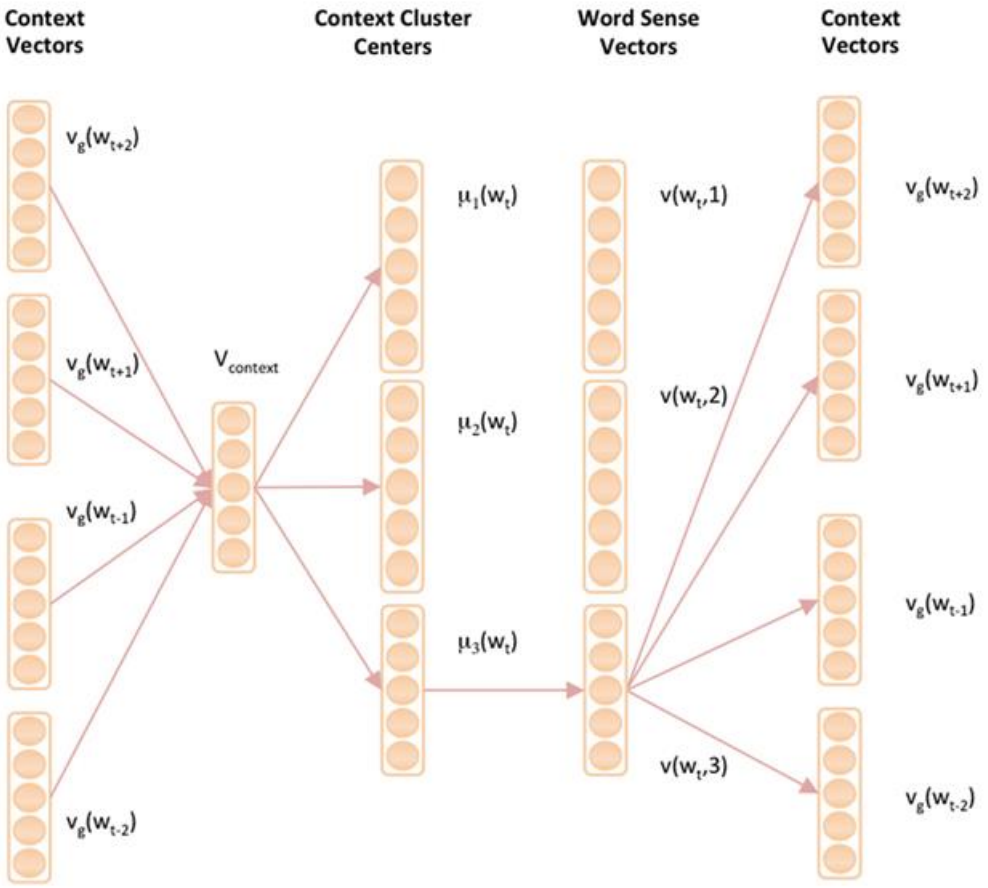


Figure 2.10 Cluster-weighted context embeddings

2.1.2.3.1 Sense2vec

Multi-sense word embedding models are more computationally expensive to train and apply in relation to single-sense models. **Sense2vec** is a simpler method to achieve world-sense disambiguation that leverages supervised labeling such as part-of-speech. It is an efficient method that eliminates the need for clustering during

training as seen in context embeddings. For instance, the meanings of the word plant are distinct based on its use as a verb or noun:

- verb: He planted the tree.
- noun: He watered the plant.

The sense2vec model can learn different word senses of this word by combining a single-sense embedding model with POS labels (Fig. 2.11). Given a corpus, sense2vec will create a new corpus for each word for each sense by concatenating a word with its POS label. The new corpus is then trained using word2vec's CBOW or skip-gram to create word embeddings that incorporate word sense (as it relates to their POS usage).

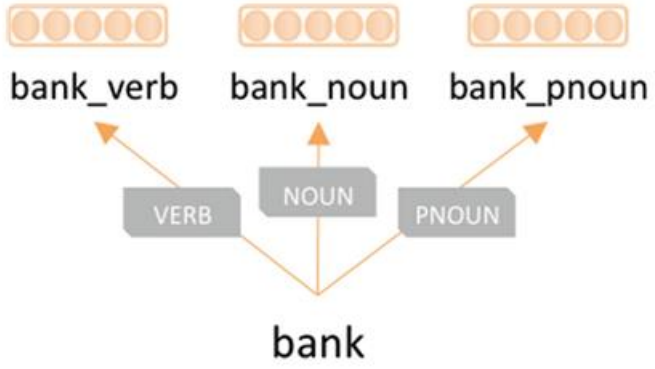


Figure 2.11 Sense2vec with POS supervised labeling

Sense2vec has been shown to be effective for many NLP tasks beyond word-sense disambiguation (Fig. 2.12).

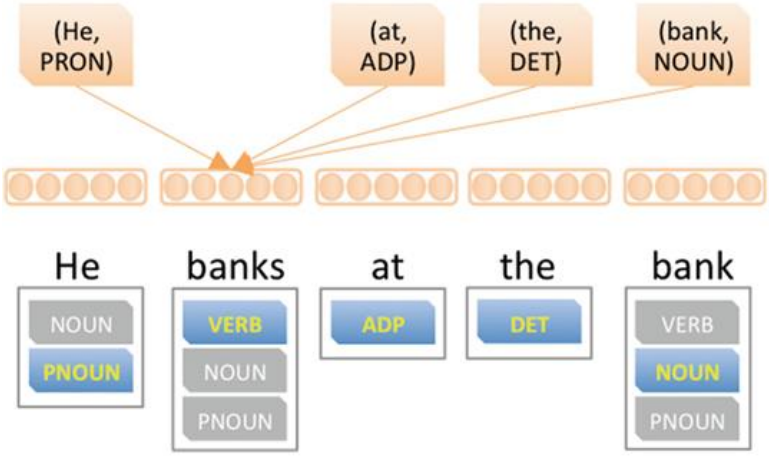


Figure 2.12 Sense2vec

2.1.2.4 Contextualized Embeddings

In the past year, a number of new methods leveraging contextualized embeddings have been proposed. These are based on the notion that embeddings for words should be based on contexts in which they are used. This context can be the position and presence of surrounding words in the sentence, paragraph, or document. By generatively pre-training contextualized embeddings and language models on massive amounts of data, it became possible to discriminatively fine-tune models on a variety of tasks and achieve state-of-the-art results. This has been commonly referred to as “NLP’s ImageNet moment”.

One of the notable methods is the **Transformer** model, an attention-based stacked encoder–decoder architecture that is pre-trained at scale. Vaswani applied this model to the task of machine translation and broke performance records.

Another important method is **ELMo**, short for Embeddings from Language Models, which generates a set of contextualized word representations that effectively capture syntax and semantics as well as polysemy. These representations are actually the internal states of a bidirectional, character-based LSTM language model that is pre-trained on a large external corpus.

Building on the power of Transformers, a method has recently been proposed called **BERT**, short for Bidirectional Encoder Representations from Transformers. BERT is a transformer-based, masked language model that is bidirectionally trained to generate deep contextualized word embeddings that capture left-to-right and right-to-left contexts. These embeddings require very little fine-tuning to excel at downstream complex tasks such as entailment or question-answering. BERT has broken multiple performance records and represents one of the bright breakthroughs in language representations today.

2.2 Automatic Speech Recognition

Automatic speech recognition (ASR) grew significantly during the last decade, and deep learning is an important part of it. In a word, ASR is the process of getting computer understandable text from a human speech (Fig. 2.13). It smoothed out the crack in human–computer communication, bringing it to the next level. Historically, ASR is tightly coupled with computational linguistics, given its close connection with natural language, and phonetics, given the variety of speech sounds that can be produced by humans.

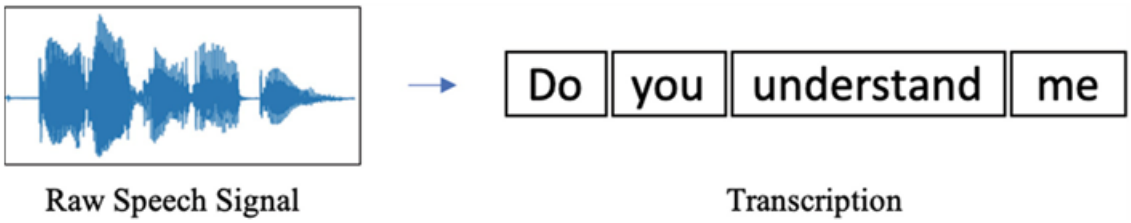


Figure 2.13

The focus of ASR is to convert a digitized speech signal into computer readable text, referred to as the transcript.

Simply put, ASR can be described as follows: given an input of audio samples X from a recorded speech signal, apply a function f to map it to a sequence of words W that represent the transcript of what was said.

$$W = f(X)$$

However, finding such a function is quite difficult, and requires consecutive modeling tasks to produce the sequence of words.

These models must be robust to variations in speakers, acoustic environments, and context. For example, human speech can have any combination of time variation (speaker speed), articulation, pronunciation, speaker volume, and vocal variations (raspy or nasally speech) and still result in the same transcript.

Linguistically, additional variables are encountered such as prosody (rising intonation when asking a question), mannerisms, spontaneous speech, also known as filler words (“um”s or “uh”s), all can imply different emotions or implications, even

though the same words are spoken. Combining these variables with any number of environmental scenarios such as audio quality, microphone distance, background noise, reverberation, and echoes exponentially increases the complexity of the recognition task.

The topic of speech recognition can include many tasks such as keyword spotting, voice commands, and speaker verification (security). Current research is concentrated on the task of speech-to-text (STT).

2.2.1 Acoustic features

The selection of acoustic features for ASR is a crucial step. Features extracted from the acoustic signal are the fundamental components for any model building as well as the most informative component for the artifacts in the acoustic signal. Thus, the acoustic features must be descriptive enough to provide useful information about the signal, as well as resilient enough to the many perturbations that can arise in the acoustic environment.

2.2.1.1 Speech Production

Let's begin with a quick overview of how humans produce speech. While a full study of the anatomy of the human vocal system is too much, some knowledge of human speech production can be helpful. The physical production of speech consists of changes in air pressure that produces compression waves that our ears interpret in conjunction with our brain. Human speech is created from the vocal tract and modulated with the tongue, teeth, and lips (often referred to as articulators):

- Air is pushed up from the lungs and vibrates the vocal cords (producing quasi-periodic sounds).
- The air flows into the pharynx, nasal, and oral cavities.
- Various articulators modulate the waves of air.
- Air escapes through the mouth and nose.

Human speech is usually limited to the range 85 Hz–8 kHz, while human hearing is in the range 20 Hz–20 kHz.

2.2.1.2 Raw Waveform

The waves of air pressure produced are converted into a voltage via a microphone and sampled with an analog-to-digital converter. The output of the recording process is a 1-dimensional array of numbers representing the discrete samples from the digital conversion. The digitized signal has three main properties: sample rate, number of channels, and precision (sometimes referred to as bit depth). The **sample rate** is the frequency at which the analog signal is sampled (in Hertz). The number of **channels** refers to audio capture with multiple microphone sources. Single-channel audio is referred to as monophonic or mono audio, while stereo refers to two-channel audio. Additional channels such as stereo and multi-channel audio can be useful for signal filtering in challenging acoustic environments [BW13]. The **precision** or **bit depth** is the number of bits per sample, corresponding to the resolution of the information.

Standard telephone audio has a sampling rate of 8 kHz and 16-bit precision. CD quality is 44.1 kHz, 16-bit precision, while contemporary speech processing focuses on 16 kHz or higher.

Sometimes bit rate is used to measure the overall quality of audio computed by:

```
bit rate = sample rate * precision * number of channels
```

The raw speech signal is high dimensional and difficult to model. Most ASR systems rely on features extracted from the audio signal to reduce the dimensionality and filter unwanted signals. Many of these features come from some form of spectral analysis that converts the audio signal to a set of features that strengthen signals that mimic the human ear. Many of these methods depend on computing a short time Fourier transform (STFT) on the audio signal using FFT, filter banks, or some combination of the two.

2.2.1.3 MFCC

Mel frequency cepstral coefficients (MFCC) are the most commonly used features for ASR. Their success relies upon their ability to perform similar types of filtering that correlates to the human auditory system and their low dimensionality.

There are seven steps to computing the MFCC features. The overall process is shown in Fig. 2.14. These steps are similar for most feature generation techniques, with some variability in the types of filters that are used and the filter banks applied. These steps are:

- Pre-emphasis
- Framing
- Hamming windowing
- Fast Fourier transform
- Mel filter bank processing
- Discrete cosine transform (DCT)
- Delta energy and delta spectrum.

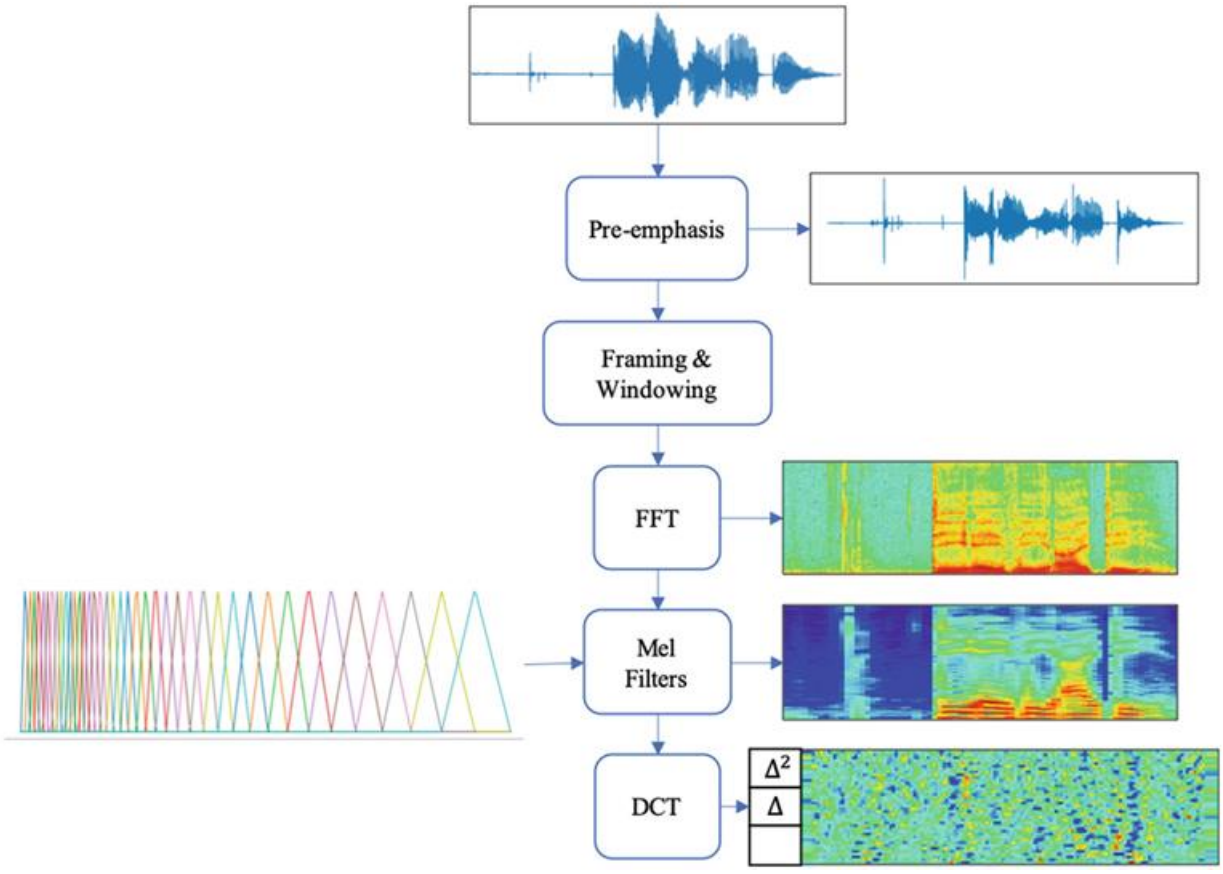


Figure 2.14 MFCC processing diagram

2.2.1.4 Other Feature Types

Many acoustic features have been proposed over the years, applying different filters and transforms to highlight various aspects of the acoustic spectrum. Many of these approaches relied on hand engineered features such as MFCCs, gammatone features, or perceptual linear predictive coefficients; however, MFCCs remain the most popular.

One of the downsides of MFCC features (or any manually engineered feature set) is the sensitivity to noise due to its dependence on the spectral form. Low dimensionality of the feature space was highly beneficial with earlier machine learning techniques, but with deep learning approaches, such as convolutional neural networks, higher resolution features can be used or even learned.

Overall MFCC features are efficient to compute, apply useful filters for ASR, and decorrelate the features. They are sometimes combined with additional speaker-specific features (typically i-vectors) to improve the robustness of the model.

2.2.1.4.1 Sense2vec

Various attempts have been tried to learn the feature representations directly, rather than relying on engineered features, which may not be best for the overall task of reducing WER. Some of the approaches include: supervised learning of features with DNNs, CNNs on raw speech for phone classification, combined CNN-DNN features, or even unsupervised learning with RBMs.

Automatically learned features improve quality in specific scenarios but can also be limiting across domains. Features produced with supervised training learn to distinguish between the examples in the dataset and may be limited in unobserved environments. With the introduction of end-to-end models for ASR, these features are tuned during the end-to-end task alleviating the two-stage training process.

2.2.2 Phones

Following from NLP, the most logical linguistic representation for transforming speech into a transcript may seem to be words, ultimately because a word-level transcript is the desired output and there is meaning attached at the word-level. Practically speaking, however, speech datasets tend to have few transcribed examples per word, making word-level modeling difficult. A shared representation for words is desirable, to obtain sufficient training data for the variety of words that are possible. For example, phonemes can be used to phonetically discretize words in a particular language. Swapping one phoneme with another changes the meaning of the word (although this may not be the case for the same phonemes in another language). For example, if the third phone in the word *sweet* [swit] is changed from [i] to [ɛ], the meaning of the whole word changes: *sweat* [swɛt].

Phonemes, themselves, tend to be too strict to use practically due to the attachment of meaning. Instead phones are used as a phonetic representation for the linguistic units (with potentially multiple phones mapping to a single phoneme). Phones do not map to any specific language, but rather, are absolute to speech itself, distinguishing sounds that signify speech. Figure 2.15 shows the phone set for English.

AA	AY	EH	HH	L	OY	T	W
AE	B	ER	IH	M	P	TH	Y
AH	CH	EY	IY	N	R	UH	Z
AO	D	F	JH	NG	S	UW	ZH
AW	DH	G	K	OW	SH	V	

Figure 2.15 English phone set (ARPAbet for ASR as used in CMU Sphinx)

With phones, words are mapped to their phonetic counterpart by using a phonetic dictionary similar to the one shown in Fig. 2.16. A phonetic entry should be present for each word in the vocabulary (sometimes more than one entry if there are multiple ways to pronounce a word). By using phones to represent words, the shared

representations can be learned from many examples across words, rather than modeling the full words.

Word	Phone Representation
a	AH
aardvark	AA R D V AA R K
aaron	EH R AH N
aarti	AA R T IY
...	...
zygote	Z AY G OW T

Figure 2.16 Phonetic dictionary for supported words in an ASR system

If every word were pronounced with the same phones, then a mapping from the audio to the set of phones to words would be a relatively straight-forward transformation. However, audio exists as a continuous stream, and a speech signal does not necessarily have defined boundaries between the phone units or even words. The signal can take many forms in the audio stream and still map to the same interpretable output. For example, the speaker's pace, accent, cadence, and environment can all play significant roles in how to map the audio stream into an output sequence. The words spoken depend not only on the phone at any given moment, but also on the states that have come before and after the context. This natural dynamic in speech places a strong emphasis on the dependency of the surrounding context and phones.

Combining phone states is a common strategy to improve quality, rather than relying on their canonical representations. Specifically, the transitions between words can be more informative than single phone states. In order to model this, **diphones** - parts of two consecutive phones, **triphones**, or extended to **senones** (triphone context-dependent units) can be used as the linguistic representation or intermediary rather than phones themselves. Many methods exist for combining the phone representations with additional context, modeling them directly or by learning a statistical hierarchy of the state combinations, and most traditional approaches rely on

these techniques.

Although ASR focuses on *recognition* rather than *interpretation* (e.g., the accuracy on recognizing spoken words rather than context-dependent word sequence modeling), the contextual understanding is an important aspect. In the case of homophones, two words with the same phonetic representation and different spellings, predicting the correct word relies entirely on the surrounding context. In this case, some of the issues can be overcome with a language model. Incorrect phonetic substitutions further complicate matters. For example, in English, the representations of *pin* [P IH N] and *pen* [P EH N] are distinct. However, although these words do have different phonetic representations, they are commonly mistakenly said interchangeably or pronounced similarly, requiring the correct selection to depend on the context more so than the phones themselves. With the inclusion of accents, phonetic representations can contain even more conflicts, requiring alternative methods to determine speaker-specific features. These types of scenarios are crucial in ASR, for there are many times that humans may say the wrong word, and yet the context and intent can still be interpreted. All of these real-world factors of spoken language contribute the complexity of automatic speech recognition in practice.

2.2.3 Statistical Speech Recognition

Statistical ASR focuses on predicting the most probable word sequence given a speech signal, via an audio file or input stream. Early approaches did not use a probabilistic focus, aiming to optimize the output word sequence by applying templates for reserved words to the input acoustic features (this was historically used for recognizing spoken digits). Dynamic time warping (DTW) was an early way to expand this templating strategy by finding the “lowest constrained path” for the templates. This approach allowed for variations in the input time sequence and output sequence; however, it was difficult to come up with appropriate constraints, such as distance metrics, how to choose templates, and the lack of a statistical, probabilistic

foundation. These drawbacks made the DTW-templating approach challenging to optimize.

A probabilistic approach was soon formed to map the acoustic signal to a word sequence. Statistical sequence recognition introduced a focus on maximum posterior probability estimation. Formally, this approach is a mapping from a sequence of acoustic, speech features, *X*, to a sequence of words, *W*. The acoustic features are a sequence of feature vectors of length *T*:

$$X = \{\boldsymbol{x}_t \in \mathbb{R}^D \mid t = 1, \dots, T\}$$

and the word sequence is defined as $W = \{w_n\, V \mid n = 1, \dots, N\}$ having a length *N*, where *V* is the vocabulary. The most probable word sequence W∗ can be estimated by maximizing P(W|X) for all possible word sequences, V∗. Probabilistically this can be written as:

$$W^* = \underset{W \epsilon V^*}{arg\max} P(W|X)$$

Solving this quantity is the center of ASR. Traditional approaches factorize this quantity, optimizing models to solve each component, whereas more recent end-to-end deep learning methods focus on optimizing for this quantity directly.
Using Bayes' theorem, statistical speech recognition is defined as:

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)}$$

The quantity *P(W)* represents the language model (the probability of a given word sequence) and *P(X|W)* represents the acoustic model. Because this equation drives the maximization of the numerator to achieve the most likely word sequence, the goal does not depend on *P(X)*, and it can be removed:

$$W^* = \underset{W \epsilon V^*}{arg\max} P(X|W)P(W)$$

An overview of statistical ASR is illustrated in Fig 2.17.

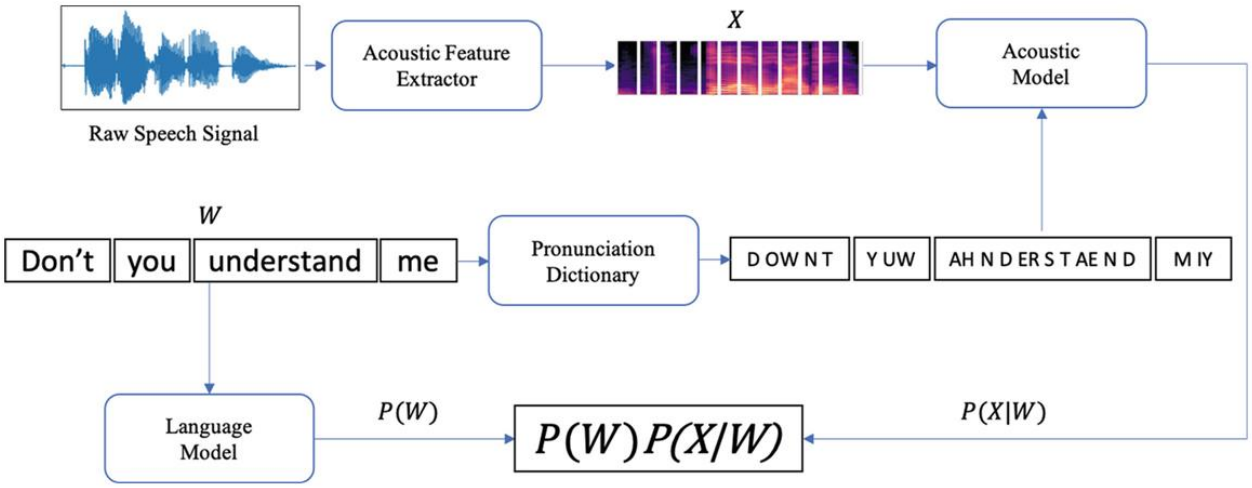


Figure 2.17 Diagram of statistical speech recognition

Often, one of the most challenging components of speech recognition is the significant difference between the number of steps in the input sequence compared to the output sequence *(T N)*. For example, extracted acoustic features may represent a 10 ms frame from the audio signal. A typical ten-word utterance could have a duration of 3-s utterance, leading to an input sequence length of 300 and a target output sequence of 10. Thus, a single word can spread many frames and take a variety of forms, as shown in Fig. 8.9. It is, therefore, sometimes beneficial to split a word into sub-components that span fewer frames.

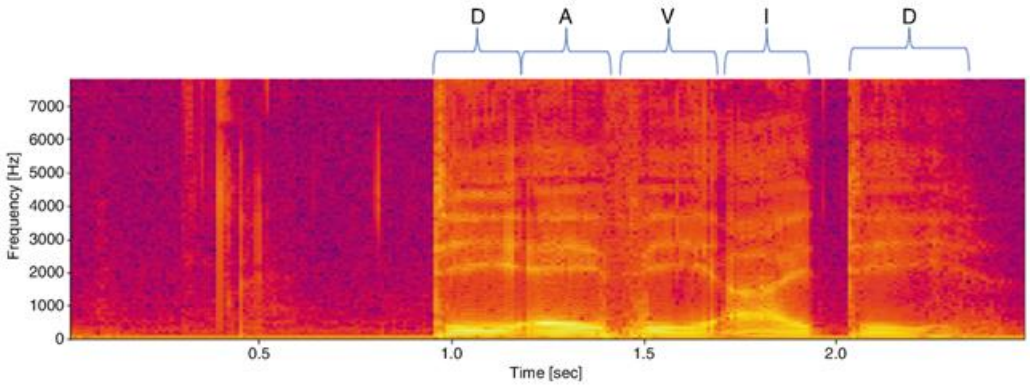


Figure 2.18 Spectrogram of a 16 kHz speech utterance, reciting the letters "D A V I D."

2.2.3.1 HMM Decoding

The decoding process for an HMM-based ASR model finds the optimal word sequence, combining the various models. The process decodes a state sequence from the acoustic features initially and then decodes to the optimal word sequence from the state sequence. Phonetic decoding has traditionally relied on interpreting the HMMs probability lattice constructed for each word from the phonetic lexicons according to the acoustic features. Decoding can be done using the Viterbi algorithm on the HMM output lattice, but this is expensive for large vocabulary tasks. Viterbi decoding performs an exact search efficiently, making it infeasible for a large vocabulary task. Beam search is often used instead to reduce the computation. The decoding process uses backtracking to keep track of the word sequence produced.

During prediction, decoding the HMM typically relies on using weighted automata and transducers. In a simple case, weighted finite state acceptors (WFSA), the automata are composed of a set of states (initial, intermediate, and final), a set of transitions between states with a label and weight, and final weights for each final state. The weights express the probability, or cost, of each transition. HMMs also can be expressed in the form of finite state automata. In this approach, a transition connects each state. WFSA accept or deny possible decoding paths depending on the states and the possible transitions. The topology could represent a word, the possible word pronunciation(s), or the probabilities of the states in the path to result in this word, (Fig. 2.19, 2.20, 2.21). Decoding, therefore, depends on combining the state models from the HMM with the pronunciation, dictionary, and n-gram language models that must be combined in some way.

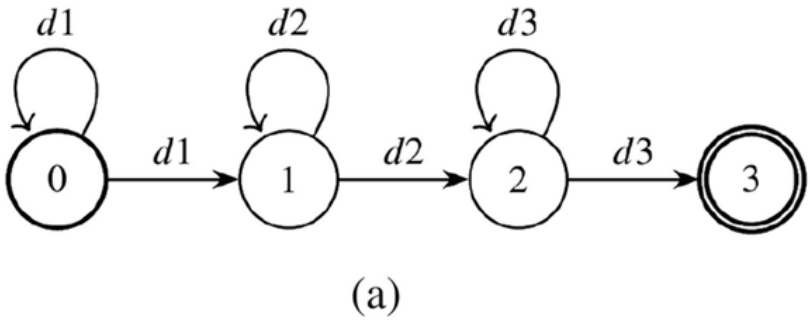
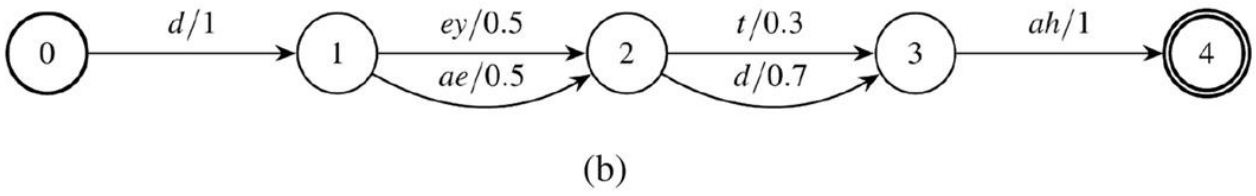
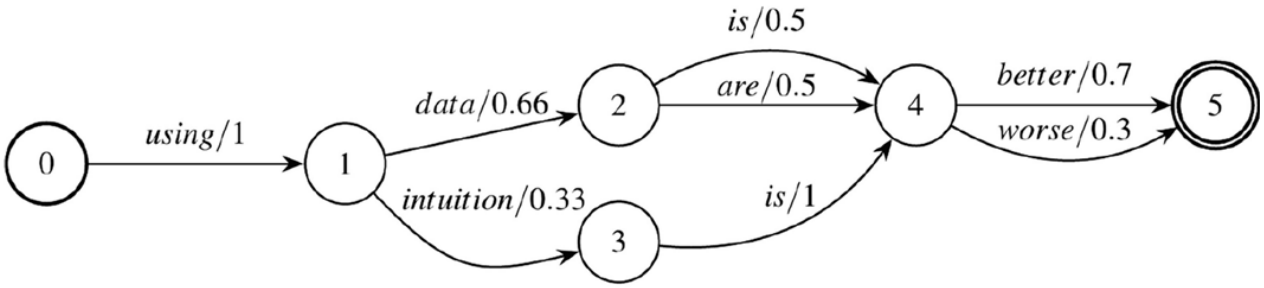


Figure 2.19 HMM state representation



(b)

Figure 2.20 Phone state transitions for the word data



(c)

Figure 2.21 Grammar state model

Usually, weighted finite state transducers (WFST) are used to represent the different levels of state transition in the decoding phase. WFSTs transduce an input sequence to an output sequence. WFSTs add an output label, which can be used to tie different levels of the decoding relationships together, such as phones and words. A WFSA is a WFST without the output label. The WFST representation allows models to be combined and optimized jointly via its structural properties with efficient algorithms: compositionality, determinism, and minimization. The composition property allows for different types of WFSTs to be constructed independently and composed together, such as combining a lexicon (phones to words) WFST and a probabilistic grammar. Determinism forces unique initial states, where no two transitions leaving a state share the same input label. Minimization combines redundant states and can be thought of as suffix sharing. Thus, the whole decoding algorithm for a DNN-HMM hybrid model can be represented by WFSTs via four transducers:

- HMM: mapping HMM states to CD phones
- Context-dependency: mapping CD phones to phones
- Pronunciation lexicon: mapping phones to words
- Word-level grammar: mapping words to words.

In Kaldi, for example, these transducers are referred to as H, C, L, and G, respectively. Compositionality allows a composition between L and G into a single transducer, L G, that maps phone sequences to a word sequence. Practically, the composition of these transducers may grow too large, so the conversion usually takes the form: *HCLG*, where

$$HCLG = min(det(H \circ min(det(C \circ min(det(L \circ G))))))$$

2.2.4 Error Metrics

The most commonly used metric for speech recognition is word error rate (WER). WER measures the edit distance between the prediction and the target by considering the number of insertions, deletions, and substitutions, using the Levenshtein distance measure.

Word error rate is defined as:

$$WER = 100 \times \frac{I + D + S}{N}$$

where

- I is the number of word insertions,
- D is the number of word deletions,
- S is the number of word substitutions, and
- N is the total number of words in the target.

For character-based models and character-based languages, error metrics focus on CER (character error rate), sometimes referred to as LER (letter error rate).

$$CER = 100 \times \frac{I + D + S}{N}$$

where

- I is the number of character insertions,
- D is the number of character deletions,
- S is the number of character substitutions, and
- N is the total number of characters in the target.

CER and WER are used to identify how closely a prediction resembles its target, giving a measurement of the overall system. They are straight-forward to compute and give a straight-forward summary of the recognition system's quality.

3. ASSISTANT IMPLEMENTATION

Nowadays plenty of companies use chat or conversational bots somehow. There are dozens of companies that have some skeletons for building AI assistants. That is why it is very reasonable to use some known basement for an own implementation.

Implementation of assistant is based on Dialogflow interface backed by custom knowledge base and additional processing with Python and related libraries (e.g. *nltk, pattern, vocabulary, spaCy, CoreNLP, TextBlob*).

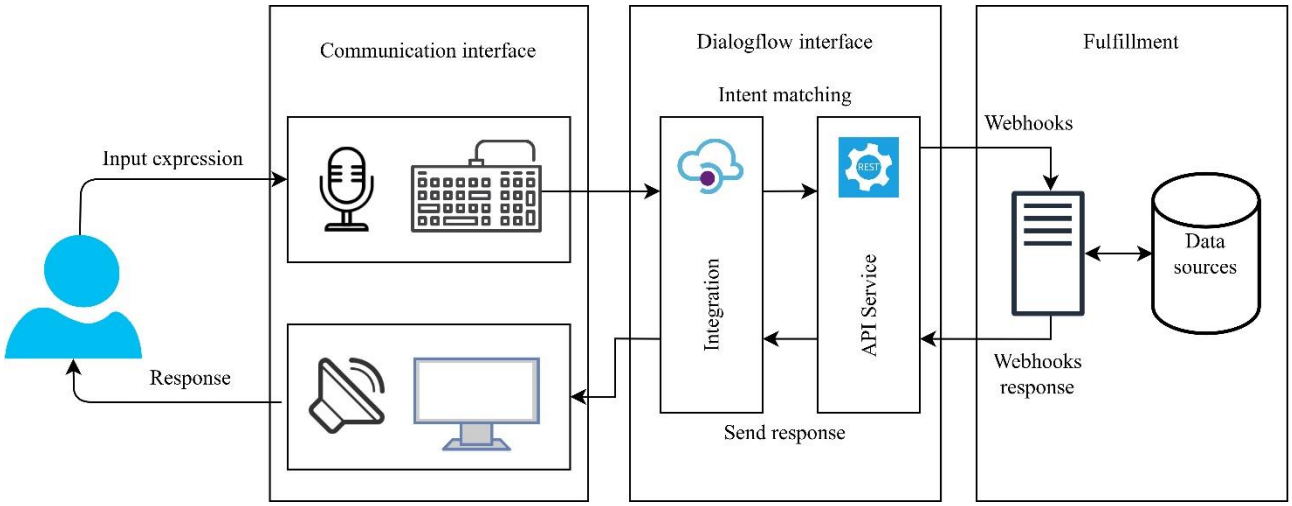


Figure 3.1 Assistant workflow

All related source code and working example can be found by the link - https://master.knu.apologist.io.

CONCLUSION

During the current work theories and concepts related to Artificial Intelligence and Natural Language Processing were put into practice. Topic of Natural Language Dialog Systems was studied in depth and it was conducted a broad review of current usages of a related technology on the language learning market.

This research aimed to identify application of natural language dialog systems in the education and implement a prototype of audio-based bot to assist in education process. The results indicate that to have truly useful assistant it is needed enormous amount of work to be done by a team of engineers. While implementing an assistant with primitive behavior and actions is a reachable task for one-man workforce.

There were reviewed language learning leaders on the market and it was indicated that modern AI technologies almost not used except some primitive tasks (e.g. spaced repetition with flash cards). That is why application of Artificial Intelligence and in particular Natural Language Processing has a very big potential in education in terms of efficiency of learning as well as in market advance for those who can successfully use it.

REFERENCES

[1] M. Abadi, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015.

[2] E. H. a. Y. S. John Duchi, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, pp. 2121-2159, 2011.

[3] Y. B. a. A. C. Ian Goodfellow, Deep Learning, MIT Press, 2016.

[4] A. G. Ivakhnenko, "The group method of data handling - a rival of the method of stochastic approximation," *Soviet Automatic Control*, pp. 43-55, 1968.

[5] I. S. a. G. E. H. Alex Krizhevsky, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[6] F. M. a. Y. Bengio, "Hierarchical Probabilistic Neural Network Language Model," *Aistats*, vol. 5, no. Citeseer, pp. 246-252, 2005.

[7] R. S. a. C. M. Jeffrey Pennington, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[8] F. Alshargi, "Concept2vec: Metrics for Evaluating Quality of Embeddings for Ontological Concepts," *CoRR abs/1803*, 2018.

[9] A. Bakarov, "A Survey of Word Embeddings Evaluation Methods," *CoRR abs/1801*, 2018.

[10] P. Bojanowski, "Enriching Word Vectors with Subword Information," *CoRR abs/1607*, 2016.

[11] J. C.-C. a. M. T. Pilehvar, "From Word to Sense Embeddings: A Survey on Vector Representations of Meaning," *CoRR abs/1805*, 2018.

[12] R. C. a. J. Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," in *Proceedings of the 25th International Conference on Machine Learning ACM*, 2008.

[13] E. Grave, "Edouard Grave," *CoRR abs/1802*, 2018.

[14] J. H. a. S. Ruder, "Universal Language Model Fine-tuning for Text Classification," *Association for Computational Linguistics*, 2018.

[15] A. Joulin, "Bag of Tricks for Efficient Text Classification," *CoRR abs/1607*, 2016.

[16] M. Lam, "Word2Bits - Quantized Word Vectors," *CoRR abs/1803*, 2018.

[17] Q. V. L. a. T. Mikolov, "Distributed Representations of Sentences and Documents," *CoRR abs/1405*, 2014.

[18] M. M. a. Y. S. Masataka Ono, "Word Embedding based Antonym Detection using Thesauri and Distributional Information," *HLT-NAACL*, pp. 984-989, 2015.

[19] T. S. a. Z. Liu, "Linking GloVe with word2vec," *CoRR abs/1411*, 2014.

[20] P. M. a. J. L. Andrew Trask, "sense2vec - A Fast and Accurate Method for Word Sense Disambiguation In Neural Word Embeddings," *CoRR abs/1511*, 2015.

[21] H. A. B. a. N. Morgan, "Connectionist speech recognition: a hybrid approach," *Springer Science & Business Media*, vol. 247, 2012.

[22] M. B. a. D. Ward, "Microphone arrays: signal processing techniques and applications," *Springer Science & Business Media*, 2013.

[23] R. J. W. a. K. W. W. Yedid Hoshen, "Speech acoustic modeling from raw multichannel waveforms," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference*, 2015.

[24] V. M. a. P. G. Andrew Cameron Morris, "From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition," in *Eighth International Conference on*

Spoken Language Processing, 2004.

[25] M. B. a. I. E. Lindasalwa Muda, "Voice recognition algorithms using Mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques," *arXiv preprint arXiv:1003.4083,* 2010.

[26] R. C. a. M. M. D. Dimitri Palaz, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," *arXiv preprint arXiv:1304.1018,* 2013.

[27] C. S. V. a. S. A. Z. Venkata Neelima Parinam, "Comparison of spectral analysis methods for automatic speech recognition," *INTERSPEECH,* pp. 3356-3360, 2013.

[28] Z. Tüske, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[29] D. Amodei, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *International Conference on Machine Learning*, 2016.

[30] D. Bahdanau, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference*, 2016.

[31] Y. He, "Streaming End-to-end Speech Recognition For Mobile Devices," in *arXiv preprint arXiv:1811.06621*, 2018.

[32] J. C. a. S. W. Takaaki Hori, "End-to-end Speech Recognition with Word-based RNN Language Models," *arXiv preprint arXiv:1808.02608,* 2018.

[33] B. M. a. C. Biemann, "Unspeech: Unsupervised Speech Context Embeddings," *arXiv preprint arXiv:1804.06775,* 2018.