Prediction of Gesture Timing and Study About Image Schema for Metaphoric Gestures

Fajrian Yunus

EDITE - ED 130

THÈSE DE DOCTORAT DE
SORBONNE UNIVERSITÉ

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique (Paris)
présentée et soutenue publiquement par

Fajrian YUNUS

le 13 décembre 2021

Prediction of Gesture Timing and Study About Image Schema for Metaphoric Gestures

Directrice de thèse: **Catherine PELACHAUD**

devant le jury composé de :

Mme. Sylvie GIBET, PR, IRISA, Université Bretagne Sud	Rapportrice
M. Alexandre PAUCHET, MdC (HDR), LITIS, INSA Rouen	Rapporteur
Mme. Catherine ACHARD, PR, ISIR, Sorbonne Université	Examinatrice
M. Brian RAVENET, MdC, LISN, Université Paris-Saclay	Examinateur
Mme. Catherine PELACHAUD, DR, ISIR, CNRS / Sorbonne Université	Directrice de Thèse
Mme. Chloé CLAVEL, PR, LTCI, IP Paris	Co-Encadrante

Thèse préparée à l'Institut Systèmes Intelligents et de Robotique (ISIR) - UMR 7222
Sorbonne Université & CNRS
BC173, 4 Place Jussieu, 75252 Paris CEDEX 05, France

As Isaac Newton and many other ancient scholars said ...
I am a mere dwarf. If I have seen further, it is by standing on the shoulder of giants.

Acknowledgement

Thank you to my bosses, *Directrice de Recherche* Catherine Pelachaud (main boss) and *Professeure* Chloé Clavel (secondary boss) for the guidance. Thank you to *Professeure* Sylvie Gibet and *Maître de Conférences (HDR)* Alexandre Pauchet for kindly acting as the reviewers for my dissertation. Thank you to *Professeure* Catherine Achard and *Maître de Conférences* Brian Ravenet for kindly acting as the examiners for my defense.

Thank you to my groupmates and ex-groupmates at ISIR, Sorbonne *Université* / CNRS for their support. Thank you for the gifts after my defense too! We might have worked on different projects, but I can see that many of you are a competent computer scientist. I'm sure many of you have a bright future as a computer scientist, wherever you decide to go.

Thank you to my collaborators at the Council of Coaches project, especially those who were directly involved on the system integration. Just like any other working adults in the world, I don't always like my job. However, working with you during the technical integration meetings, both at the *Universiteit* Twente and online, left me with a certain unmistakable impression: you are competent and serious computer scientists! Working with you was definitely both an honor and a pleasure.

Thank you to my ex-boss and my ex-groupmates who were/are the School of Computing, National University of Singapore (and to the visitors of the group too). Even after I left you to do PhD, from time to time I still send you an email or a message asking computer science questions, and you do actually reply. The school administration likes to talk about the school's ranking, which I remember to be consistently among or around the top-20 computer science departments in the world. But in you, I see what makes such a top-tier computer science research department. Not only that you are (very strong) former colleagues, but you are also friends.

Finally, thank you to the European Commission which funded my PhD through its European Union's Horizon 2020 research and innovation program under grant agreement number 769553 (i.e. the Council of Coaches project). This dissertation only reflects the authors' views and the European Commission is not responsible for any use that may be made of the information this dissertation contains.

Summary (English)

Human-Computer Interaction (HCI) is a domain within computer science research. However, Human-Computer Interaction has a different motivating spirit from the "mainstream" computer science research domains such as machine learning, database, or algorithm. These "mainstream" domains concern themselves to make computer more powerful, to enable computer to achieve technical feats which previously could not be done. For example, advances in database domain enables Google to search from its vast data within a few seconds. On the other hand, Human-Computer Interaction concerns itself to make computer more accessible to regular people.

One important characteristic of the modern computer is that it can show a moving graphic and also plays a sound. Therefore, it is technically possible to show a sentient-looking figure which speaks and shows non-verbal behaviors. This technology is called "Embodied Conversational Agent" (ECA). The non-verbal behaviors themselves also have practical benefits. For example, non-verbal behaviors help us to produce speech, to formulate our thoughts, and to communicate our feelings. By being able to produce and respond properly to non-verbal behaviors, Embodied Conversational Agent enables human to interact with a computer in a mode of communication which we have been using since our species appeared on Earth: face to face communication.

Non-verbal behavior has many components, such as gaze directions, facial expressions, or communicative gestures. These non-verbal behaviors are not random. For example, if we say "the elevator is going up", the gesture we probably perform is pointing upward. This gesture makes the message clearer. From this example, it can be seen that proper non-verbal behaviors serve useful functions. Therefore, having an Embodied Conversational Agent (ECA) which can generate and perceive non-verbal behaviors properly is a desirable goal. The focus of this dissertation is the generation of communicative gestures.

Generated communicative gestures have several desirable properties. The gestures should be coherent with the speech, the generated gestures should connect with each other smoothly, and the generated gestures should be diverse. Our focus is on the coherence property. We deal with two aspects of the coherence. The first aspect is the coherence between the rhythm of the speech and the gesture timing. The second aspect is the coherence between the speech semantics and the gesture shape.

There are many important developments in the past many years in this gesture generation problem. The first notable development is the shift toward machine learning. The earliest techniques to generate gestures are rule-based. However, due to the eventual complexity of the rules, researchers turn to machine learning. We also use machine learning in our work. The second notable development is the use of acoustic features (e.g. fundamental frequency, intensity, etc.). These features are extracted from the voice. This development is related to the use of machine learning, because machine learning enables processing of complex data, which is difficult at a rule-based system. We also use acoustic features extracted from the speech in our work. The third development is the recognition of the many-to-many relationship of gestures. This means that for a particular speech, there are many different compatible gestures. Some prior works use adversarial learning

to achieve this many-to-many relationship. We recognize the many-to-many relationship to a limited extent. The fourth development is the use of word embedding to represent text. Word embedding is a class of techniques which maps words into vectors in such a way that two similar words are mapped to two nearby vectors. The fifth and last development we would like to highlight is the use of image schema to represent the semantics for metaphoric gesture generation. Image schema is a recurrent pattern of reasoning which maps one entity into another. For metaphoric gesture generation, it is to map an abstract concept into a concrete object such that the object has a physical shape which can be depicted in a gesture. As a part of our work, we develop a technique to compute the embedding of image schemas.

We use a third party corpus called Gest-IS corpus for our work. The corpus consists of 9 dialogues of a dyad, a man and a woman, both of them are sitting, discussing various topics in a face-to-face setting. The corpus comes with the transcripts and the annotations of communicative or non-communicative gestures, the gesture types, and the gesture phases. However, the corpus also comes with the videos of the conversations and the corresponding audio files. The videos show the torso, hands, and face of the dyad. We use a third party software called OpenFace to extract the eyebrow movements. This can be seen as adding another annotation to the corpus. We also extract the acoustic features from the conversation audio files. The acoustic feature extraction is done by using another third party software called OpenSmile. On top of that, we also manually annotate the shape of metaphoric gestures in this corpus. This is possible because the corpus already contains the annotation of metaphoric gesture stroke timing and the videos showing the gesture being performed. However, in order to do the annotation, we define first the annotation scheme. This annotation scheme should be large enough to capture various gesture shapes, but also compact enough such that it is practical enough to be encoded. After we create the annotation scheme, we annotate the metaphoric gesture shapes accordingly. Then, we calculate and analyze the statistics of the different dimensions of the gesture shapes. We also calculate and analyze the statistics after breaking them down per image schema.

We develop a model based on recurrent neural network model with attention mechanism to predict the gesture timing. The problem is expressed as a time series prediction problem where the input is a sequence of acoustic features of the speech and the output is a sequence of gesture classes. The sequence of gesture classes represents the gesture timing. The neural network is trained, validated, and tested with the data from the aforementioned corpus. Besides that, we also develop an objective evaluation measure which tolerates shift and dilation to some extent. This is done in the spirit of recognizing many-to-many relationship between speech and gestures. We also experiment with including eye brow movements which we extracted from the corpus for the prediction. We also experiment with training and validating the model with the data of one speaker only and testing the model on the data of the other speaker. Finally, we do a subjective experiment to evaluate the naturalness, time consistency, and semantic consistency of the output of the model.

In our another contribution, we make some improvements to Ravenet et al's algorithm which compute gestures from a free-form text input via image schema. There are two improvements we make. The first improvement is replacing the word sense disambiguation technique employed in the Ravenet et al's algorithm. The second improvement is that we add more edge types as the permissible path for the WordNet graph traversal.

For our last contribution, we develop a method to represent image schemas as vectors. The method is based on BERT and SenseBERT word embedding techniques to convert a free-form text into vectors. We also compute the corresponding image schemas from the aforementioned free-form text input by using the improved Ravenet et al's algorithm. Based on that, we measure the clustering behavior of the vectors which belong to the same image schema. We then define the centroid of each cluster as the vector representation of the image schema. However, with the image schemas being represented as vectors, it also becomes possible to calculate the distances between them, which are a proxy of similarities between different image schemas. Therefore, we also measure the distances between different image schemas to find out which image schemas are close/similar to each other. Finally, we display some visualizations to show the relative distances between different image schemas.

Keywords : machine learning, neural network, gesture, recurrent neural network, long short term memory, attention, word embedding, BERT, SenseBERT, cluster, image schema

Sommaire (Français)

L'Interaction Homme-Machine (« Human-Computer Interaction » ou HCI en anglais) est un domaine de la recherche en informatique. Cependant, l'Interaction Homme-Machine a un esprit de motivation différent des domaines de recherche en informatique “traditionnels” tels que l'apprentissage automatique, les bases de données ou les algorithmes. Ces domaines « mainstream » se préoccupent eux-mêmes de rendre l'ordinateur plus puissant, de permettre à l'ordinateur de réaliser des prouesses techniques qui n'étaient pas réalisables auparavant. Par exemple, les progrès dans le domaine des bases de données permettent à Google d'effectuer des recherches à partir de ses vastes données en quelques secondes. D'autre part, l'Interaction Homme-Machine se préoccupe de rendre l'ordinateur plus accessible aux gens ordinaires.

Une caractéristique importante de l'ordinateur moderne est qu'il peut afficher un graphique en mouvement et jouer également un son. Par conséquent, il est techniquement possible de montrer une figure apparemment sensible qui parle et montre des comportements non verbaux. Cette technologie s'appelle l'Agent Conversationnel Animé (« Embodied Conversational Agent » ou ECA en anglais). Les comportements non verbaux eux-mêmes ont également des avantages pratiques. Par exemple, les comportements non verbaux nous aident à produire la parole, à formuler nos pensées et à communiquer nos sentiments. En étant capable de produire et de répondre correctement aux comportements non verbaux, un Agent Conversationnel Animé permet à l'homme d'interagir avec un ordinateur dans un mode de communication que nous utilisons depuis que notre espèce est apparue sur Terre : la communication face à face.

Le comportement non verbal a de nombreuses composantes, telles que les directions du regard, les expressions faciales ou les gestes communicatifs. Ces comportements non verbaux ne sont pas aléatoires. Par exemple, si nous disons “l'ascenseur monte”, le geste que nous effectuons probablement pointe vers le haut. Ce geste rend le message plus clair. À partir de cet exemple, on peut voir que les comportements non verbaux appropriés remplissent des fonctions utiles. Par conséquent, avoir un Agent Conversationnel Animé capable de générer et de percevoir correctement des comportements non verbaux est un objectif souhaitable. L'objet de cette thèse est la génération de gestes communicatifs.

Les gestes communicatifs générés ont plusieurs propriétés souhaitables. Les gestes doivent être cohérents avec le discours, les gestes générés doivent se connecter les uns aux autres en douceur et les gestes générés doivent être diversifiés. Nous nous intéressons à la propriété de cohérence. Nous traitons deux aspects de la cohérence. Le premier aspect est la cohérence entre le rythme de la parole et le chronométrage du geste. Le deuxième aspect est la cohérence entre la sémantique de la parole et la forme du geste.

Il y a eu de nombreux développements importants au cours des dernières années dans ce problème de génération de gestes. Le premier développement notable est le passage à l'apprentissage automatique. Les premières techniques pour générer des gestes sont basées sur des règles. Cependant, en raison de la complexité éventuelle des règles, les chercheurs se tournent vers l'apprentissage automatique. Nous utilisons également l'apprentissage au-

tomatique dans notre travail. La deuxième évolution notable est l'utilisation de caractéristiques acoustiques (par exemple, fréquence fondamentale, intensité, etc.). Ces caractéristiques sont extraites de la voix. Ce développement est lié à l'utilisation de l'apprentissage automatique, car l'apprentissage automatique permet le traitement de données complexes, ce qui est difficile dans un système basé sur des règles. Nous utilisons également des caractéristiques acoustiques extraites de la parole dans notre travail. Le troisième développement est la reconnaissance de la relation plusieurs-à-plusieurs des gestes. Cela signifie que pour un discours particulier, il existe de nombreux gestes compatibles différents. Certains travaux antérieurs utilisent l'apprentissage antagoniste pour réaliser cette relation plusieurs-à-plusieurs. Nous reconnaissons la relation plusieurs à plusieurs dans une mesure limitée. Le quatrième développement est l'utilisation du plongement de mots pour représenter le texte. Le plongement de mots est une classe de techniques qui mappent des mots dans des vecteurs de telle manière que deux mots similaires sont mappés sur deux vecteurs proches. Le cinquième et dernier développement que nous voudrions souligner est l'utilisation du schéma d'image pour représenter la sémantique pour la génération de gestes métaphoriques. Le schéma d'image est un schéma récurrent de raisonnement qui associe une entité à une autre. Pour la génération de gestes métaphoriques, il s'agit de mapper un concept abstrait dans un objet concret tel que l'objet ait une forme physique qui peut être représentée dans un geste. Dans le cadre de notre travail, nous développons une technique pour calculer le plongement de schémas d'images.

Nous utilisons un corpus tiers qui s'appelé corpus Gest-IS pour notre travail. Le corpus se compose de 9 dialogues d'une dyade, un homme et une femme, tous deux assis, discutant de divers sujets en face-à-face. Le corpus est accompagné des retranscriptions et des annotations des gestes communicants ou non communicants, des types de gestes et des phases gestuelles. Cependant, le corpus est également accompagné des vidéos des conversations et des fichiers audio correspondants. Les vidéos montrent le torse, les mains et le visage de la dyade. Nous utilisons un logiciel tiers qui s'appelle OpenFace pour extraire les mouvements des sourcils. Cela peut être vu comme l'ajout d'une autre annotation au corpus. Nous extrayons également les caractéristiques acoustiques des fichiers audio de conversation. L'extraction des caractéristiques acoustiques est effectuée à l'aide d'un autre logiciel tiers qui s'appelle OpenSmile. En plus de cela, nous annotons également manuellement la forme des gestes métaphoriques dans ce corpus. Ceci est possible car le corpus contient déjà l'annotation du chronométrage métaphorique des gestes et les vidéos montrant le geste en cours d'exécution. Cependant, afin de faire l'annotation, nous définissons d'abord le schéma d'annotation. Ce schéma d'annotation doit être suffisamment grand pour capturer diverses formes de gestes, mais également suffisamment compact pour être suffisamment pratique pour être encodé. Après avoir créé le schéma d'annotation, nous annotons les formes de gestes métaphoriques en conséquence. Ensuite, nous calculons et analysons les statistiques des différentes dimensions des formes gestuelles. Nous calculons et analysons également les statistiques après les avoir décomposées par schéma d'image.

Nous développons un modèle basé sur un modèle de réseau neuronal récurrent avec un mécanisme d'attention pour prédire le chronométrage des gestes. Le problème est exprimé sous la forme d'un problème de prédiction de séries chronologiques où l'entrée est une séquence de caractéristiques acoustiques de la parole et la sortie est une séquence de classes de gestes. La séquence de classes de gestes représente la synchronisation des gestes. Le réseau de neurones est formé, validé et testé avec les données du corpus susmentionné. En plus de cela, nous développons également une mesure d'évaluation objective qui tolère le décalage et la dilatation dans une certaine mesure. Ceci est fait dans

l'esprit de reconnaître la relation plusieurs-à-plusieurs entre la parole et les gestes. Nous expérimentons également l'inclusion des mouvements des sourcils que nous avons extraits du corpus pour la prédiction. Nous expérimentons également l'entraînement et la validation du modèle avec les données d'un seul locuteur et testons le modèle sur les données de l'autre locuteur. Enfin, nous réalisons une expérience subjective pour évaluer le caractère naturel, la cohérence temporelle et la cohérence sémantique de la sortie du modèle.

Dans notre autre contribution, nous apportons quelques améliorations à l'algorithme de Ravenet et al qui calcule les gestes à partir d'une entrée de texte de forme libre via un schéma d'image. Nous apportons deux améliorations. La première amélioration consiste à remplacer la technique de désambiguïsation lexicale employée dans l'algorithme de Ravenet et al. La deuxième amélioration est que nous ajoutons plus de types d'arêtes comme chemin autorisé pour la traversée du graphe WordNet.

Pour notre dernière contribution, nous développons une méthode pour représenter les schémas d'images sous forme de vecteurs. La méthode est basée sur les techniques de plongement de mots BERT et SenseBERT pour convertir un texte de forme libre en vecteurs. Nous calculons également les schémas d'image correspondants à partir de l'entrée de texte de forme libre susmentionnée en utilisant l'algorithme amélioré de Ravenet et al. Sur cette base, nous mesurons le comportement de partitionnement des vecteurs qui appartiennent au même schéma d'image. Nous définissons ensuite le centroïde de chaque cluster comme la représentation vectorielle du schéma d'image. Cependant, les schémas d'images étant représentés sous forme de vecteurs, il devient également possible de calculer les distances entre eux, qui sont un proxy des similitudes entre différents schémas d'images. Par conséquent, nous mesurons également les distances entre différents schémas d'images pour déterminer quels schémas d'images sont proches/similaires les uns aux autres. Enfin, nous affichons quelques visualisations pour montrer les distances relatives entre les différents schémas d'image.

Abstract

"Embodied Conversational Agent" (ECA) are virtual agents endowed with the capacity to communicate verbally and non-verbally. Non-verbal behavior has many components, such as gaze directions, facial expressions, or communicative gestures. These non-verbal behaviors are not random. For example, if we say "the elevator is going up", the gesture we probably perform is pointing upward. The focus of this dissertation is the generation of communicative gestures for Embodied Conversational Agent.

We use a third party corpus called Gest-IS corpus for our work. The corpus comes with the transcripts and the annotations of communicative or non-communicative gestures, the gesture types, and the gesture phases. The corpus also has the videos of the conversations and the corresponding conversation audio files. From those videos, we extract the eyebrow movements. From the audio files, we extract the acoustic features. These eyebrow movements and the acoustic features can be seen as additional annotations. On top of that, we also manually annotate the shape of metaphoric gestures in this corpus. In order to do the gesture shape annotation, we define first the annotation scheme. With that, we annotate the metaphoric gesture shapes accordingly. We also calculate and analyze the statistics of the different dimensions of the gesture shapes.

We develop a model based on recurrent neural network model with attention mechanism to predict the gesture timing. Besides that, we also develop an objective evaluation measure which tolerates shift and dilation to some extent. This is done in the spirit of recognizing many-to-many relationship between speech and gestures. We also experiment with including eye brow movements which we extracted from the corpus for the prediction. We also experiment with training and validating the model with the data of one speaker only and testing the model on the data of the other speaker. Finally, we do a subjective experiment to evaluate the naturalness, time consistency, and semantic consistency of the output of the model.

In our another contribution, we make some improvements to Ravenet et al's algorithm which compute gestures from a free-form text input via image schema. First, we replace the word sense disambiguation technique employed in the Ravenet et al's algorithm. Secondly, we add more edge types for the WordNet graph traversal. For our last contribution, we develop a method to represent image schemas as vectors. The method works by calculating the centroid of the clusters of the word embedding vectors. The word embedding vectors themselves come from either BERT or SenseBERT word embedding techniques. The centroid of the cluster is considered as the vector representation of the image schema. With the image schemas represented as vectors, we also measure the relative distances between the image schemas. These distances are a proxy of the similarities/differences between the different image schemas.

Keywords: machine learning, neural network, gesture, recurrent neural network, long short term memory, attention, word embedding, BERT, SenseBERT, cluster, image schema

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this dissertation, we discuss our work about the generation of communicative gestures. We open this chapter by explaining the wider context behind the problem of communicative gesture generation. We start with the overview of Human-Computer Interaction and Embodied Conversational Agent and the motivating spirit of the research in this domain. After that, we explain the properties we desire from generated gestures and the important developments in the gesture generation research. Finally, we explain our contributions and we close this chapter with the organization of this dissertation.

1.1 Human-Computer Interaction and Embodied Conversational Agent (ECA)

In the year 2021, communication can be done in so many ways. We can send an e-mail, we can send an SMS, we can do a video call, we can do a phone call, and we can also talk face to face. In the 1990s and early 2000s when internet was already widespread but high-speed internet was not, mobile phones played a big part in our communication. We called other people and sent SMS to other people. Meanwhile, for long messages, we used e-mail. Turn the clock back several decades earlier to the mid-20th century, fixed-line phones were already widespread, which enabled us to talk to someone faraway relatively easily. However, written communication was still difficult. Telegram was widely-available and was relatively fast, but it was quite expensive. Therefore, detailed written communication had to be done by a letter written on a paper. Turn the clock back by one century, in the 19th century, the rising literacy among the common people and the provisioning of relatively affordable postal services enabled regular people in faraway places to communicate with each other reliably with mails. However, before that, the only way regular people could communicate with each other was by face to face communication. That means, from the time our species appeared on Earth around 200,000 years ago until the 18th or the 19th century, the only way regular people could communicate with each other was by a face to face talk. This is how most human communicate for much of the history of our species.

Written communication appeared only a few thousand years ago when human invented writing systems. However, writing media (e.g. paper, papyrus, etc.) only became affordable and widely available several hundred years ago. Meanwhile, literacy itself only became widespread in Europe and North America in the 18th and 19th century. In some

parts of the world, widespread literacy became a reality only in the late 20th century. That means, for most of our history as a species, written communication was either unavailable or was restricted to the societal elites only.

Long distance spoken communication became technically possible when telephone was invented in the late 19th century. However, telephones require an expensive communication infrastructure. In developed countries, fixed-line telephones became widespread only around the mid-20th century. In many developing countries, it became a reality only in the late 20th century. Thus, for many decades of the history of long-range spoken communication, it was available only to the societal elites.

Human-Computer Interaction, as a research domain, has a different motivating spirit from "mainstream" computer science research domains. Machine learning research is done to enable computer to spot patterns from data. Database research is done to enable computer to store and process a large amount of data efficiently. Algorithm research is done to discover ways to solve fundamental computer science problems efficiently such that the solution is general enough for all kinds of application. These "mainstream" computer science research domains concern themselves with technical challenges. Their motivating spirit is to make computer more powerful, to enable computer to achieve feats which previously could not be done. However, Human-Computer Interaction, as a research domain does not aim to make computer more powerful. Instead, their motivating spirit is to make computer more accessible to regular people.

The oldest way of using computer is through the command line. Many programs which are meant to be used only by specialists are still primarily accessible through command line. Command line can indeed be efficient at the hand of an expert. However, command line is inaccessible by the non-experts. Later on, Graphical User Interface was created and made computer more accessible to regular people. However, computer still can be made more accessible.

Unlike the earlier media of communication, such as phone, SMS, telegram, or letter, computer can show moving graphic (and also outputs sound). That means, it is technically possible to show a seemingly-sentient figure which both speaks and display non-verbal behaviors like eye gaze, gestures, and facial expressions (at least, in the 2-dimensional form). Besides that, non-verbal behaviors also have practical benefits. Among others, non-verbal behaviors help to produce speech, to formulate our thoughts, and to communicate our feelings (Pelachaud (2009)). Effectively, this technology enables the user to interact with a computer in a mode of communication which we have been using since our species appeared on Earth: face to face communication. We call such technology "Embodied Conversational Agent" (ECA). Precisely, Embodied Conversational Agent is a virtual agent endowed with the capacity to communicate both verbally and non-verbally (Cassell et al. (2004)).

1.2 Non-Verbal Communication

Non-verbal communication has many components, such as gaze directions, facial expressions, or communicative gestures. These non-verbal behaviors are not random. For example, during a conversation, we spend much of the time looking at the speaker or the objects related to the speech. Facial expressions are another useful non-verbal behavior. A speaker can use his facial expressions to convey the message more effectively. For example, he can convey friendliness with a smile. Gestures are also useful communication tools. For example, saying "the elevator is going up" while pointing above makes the message

clearer. From these examples, it can be seen that proper non-verbal behaviors do serve useful functions. Therefore, having an Embodied Conversational Agent (ECA) which can generate and perceive non-verbal behaviors properly is a desirable goal. This is indeed an active and broad research area. This dissertation focuses on the question of the generation of communicative gestures.

1.3 Desired Properties of Generated Communicative Gestures

Researchers who work on communicative gesture generation have identified several properties which should be satisfied by generated gestures. However, these properties must not be treated as the canonical list. Future researchers might identify new properties. Besides that, simply satisfying all these properties cannot be considered as a guarantee that a gesture generator is perfect.

1.3.1 Match/Coherence With The Speech

One component of the match/coherence is that the most prominent part of the gesture should happen around the pitch accent of the speech (Wagner et al. (2014)). However, the rhythm match is not the only thing which matters. The other component of the match is that the gesture's shape should match the meaning conveyed by the speech. For example, we can expect that the gesture accompanying an utterance “turn right” should indicate the right direction.

It might be tempting to think that because it is possible to express ideas in detail by writing alone (e.g. a book written by an expert writer), then verbal communication is always more expressive than gesture, and therefore it is possible to “translate” speech into gesture. However, in the natural speech like in a face to face talk, it is not always the case. For example, in a face to face talk, we might simply say “go there” while pointing to the right instead of specifically saying “go to the direction which is on my right side from my point of view”. In this case, the information about the destination is conveyed by the gesture; it cannot be inferred from the speech. Therefore, it is not always possible to “translate” the speech into a gesture.

The notion of match is made even more complicated by the fact that gesture is different from pantomime. We do not gesture all the time. Thus, it is possible to utter “turn right” without doing any gesture.

1.3.2 Smoothness

We do many gestures. There are also times when we do not do gesture. However, the transition between them should be smooth and seamless. Hasegawa et al. (2018) and Kucherenko et al. (2019) explicitly take into account the movement smoothness in their objective evaluation. Specifically, they measure the average jerk of the movement in their objective study where jerk is the temporal differentiation of the acceleration.

1.3.3 Diversity

Gesture is idiosyncratic (McNeill (1992)), which practically means that there are many possible gestures to accompany a certain utterance. For example, if the utterance is “turn right”, then it is possible that the gesture is pointing to the right by using the right hand's

thumb, or pointing to the right by using the left hand's index finger, or by both hands acting as if turning steering wheel clockwise (i.e. the movement of turning a car to the right), or by the head tilting to the right. It can be seen from this example that the relationship between the speech and the gesture is many-to-many. Wu et al. (2021) measure the distribution of the gesture output in their objective study to find out if the diversity is comparable to the one in the ground truth.

1.4 Important Developments

In this section, we explain the important developments in the gesture generation problem. We also mention the relevant landmark papers which open new lines of inquiry.

1.4.1 Machine Learning

The earliest techniques to automatically generate gestures are rule-based (see Section 3.1). Some of them have the rules set according to the knowledge from the literature alone while some others also enrich their rules with findings from statistical analysis. However, those rules eventually turn out to be complex, and therefore researchers turn to machine-learning based techniques.

1.4.2 Acoustic Features

A beneficial side effect of the shift toward machine learning is that it becomes possible to learn directly from the acoustic data. Among the rule-based systems, Cerebella (Lhommet and Marsella (2013)) is the only one which takes the acoustic as an input. Even so, the acoustic input is limited to a few discrete features. On the other hand, machine-learning based techniques can learn from real-valued acoustic features (e.g. fundamental frequency and intensity). This trend of learning from real-valued acoustic inputs was started by Levine et al. (2010) and is still done in recent works. Many machine-learning-based techniques rely on these inputs. One important supporting factor in this development is that there are programs which can extract acoustic features, such as OpenSmile (Eyben et al. (2010)). Besides that, these acoustic features are in the form of real-valued array. This is the form of input which machine learning works with.

1.4.3 Body Joint Coordinates

Another side effect of the shift toward machine learning is the way the gesture is represented. The gesture representation also moves into the lower level. The gestures are represented as 3-dimensional coordinates of the joints. Their movements are represented as a time series of those three-dimensional joint coordinates. Thus, the gestures are represented as a three-dimensional array $Y_{N,3,T}$ where T is the number of time-steps and N is the number of joints. Thus, $Y_{n,:,t}$ are the three-dimensional coordinates of the joint n at time-step t. This representation is practical for machine learning because this representation is in the form of real-valued array, which is what machine learning works with. Another supporting factor is that this data can be obtained by using a Motion Capture (MoCap) tool (see Figure 1.1 for an example). However, the limitation is that this tool is expensive and requires a special facility dedicated to it. Besides that, human actors/actresses are still needed to have their motion being captured. Thus, getting access to the

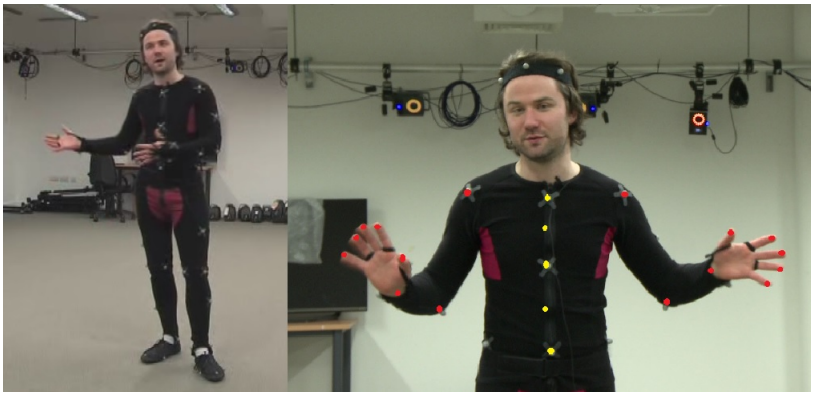

Figure 1.1 – Motion capture from Ferstl et al. (2019). The actor's body is fitted with sensors. The cameras behind the actors are parts of the Motion Capture tool

data is still a burden. Interestingly, Ginosar et al. (2019) circumvent this problem by working with two-dimensional data instead of three-dimensional data. The joint coordinates are extracted from publicly accessible videos by using OpenPose (Cao et al. (2019)). This approach is sensible because the output is to be played on a computer screen, which is only two-dimensional. Habibie et al. (2021) extend this technique further by projecting the three-dimensional coordinates from the two-dimensional coordinates.

1.4.4 Gesture Movement Smoothing

One problem which was discovered early is that it is not always obvious on how to ensure the movement is smooth. Because of this, Levine et al. (2010) limit the choices of motion segment so that the resulting movement is smooth. Subsequently, Chiu and Marsella (2014) and Bozkurt et al. (2016) solve this problem by automatically smoothen the transitions.

The movement of the gestures or between gestures should be smooth. However, the way to satisfy this property is not always obvious. Levine et al. (2010), a relatively early machine-learning-based gesture generator, have to constrain the choices of motion segment to ensure that the resulting movement is smooth. This limitation is solved by the subsequent works. Chiu and Marsella (2014) and Bozkurt et al. (2016) subsequently solve this problem by automatically smoothen the transitions.

1.4.5 Many-To-Many Relationship of Gestures

Gesture is idiosyncratic (McNeill (1992)), which practically means that there are many possible gestures to accompany a certain utterance. Thus, the relationship between utterances and gestures should be many-to-many. In a rule-based gesture generator, it can be achieved by setting several possible gestures for each scenario and assigning probability to each of those gestures. However, the recent gesture generators use neural network to generate the gestures (see Sections 3.4, 3.5, and 3.6). Neural network is stochastic during the learning process, but is deterministic after the learning process is completed. Thus, if given the same input twice, the two outputs will be the same as well. Ginosar et al. (2019) solve this problem by using adversarial learning. This is possible because adversarial learning is stochastic even after the training.

1.4.6 Word Embedding

A curious case of development is the use of word embedding (see Section 6.3.2) by Ahuja and Morency (2019) to represent the text input. Word embedding represents a word as a real-valued vector such that two similar words are mapped to two nearby vectors, although different word embedding techniques have different notions on what it means that two words are similar. Previously, the text input has to be processed first to get various linguistic features. Besides that, real-valued vectors yielded by word embedding techniques are convenient to use with machine learning. The use word embedding to represent textual input is subsequently followed by Kucherenko et al. (2020).

1.4.7 Image Schema

An orthogonal but related development to the use of word embedding to represent the textual input is the use of image schema (see Section 2.6) by Ravenet et al. (2018a,b) to represent the semantics for metaphoric gesture generation. Image schema itself is a recurrent pattern of reasoning which maps one entity into another. Unlike word embedding which represents the semantics as a vector, image schema represents the semantics as a class. However, the context where Ravenet et al. (2018a,b) use image schema is not a machine-learning-based system.

1.4.8 Multimodal Machine Learning

The last development we would like to highlight is the use of multimodal machine learning by Kucherenko et al. (2020) to use both the acoustic features and the textual features as inputs (see Section 3.6). Having both modalities as the inputs opens the possibilities to exploit richer information from both modalities. However, their experiment results suggest that simply having both modalities as inputs does not guarantee effective use of both modalities. The way to effectively use both modalities is still an open problem.

1.5 Our Contributions

We extract additional data from the corpus, namely the acoustic features and eyebrow movements. We also create a gesture shape annotation scheme and we show the statistics of the gestures according to their shape.

We develop a model to predict gesture timing. This work is a step towards generating gestures with the desired property that the gestures should match/be coherent with the speech (see Section 1.3.1), especially toward generating gestures whose timing matches the speech's rhythm. In this chapter, we also propose a measurement method for the objective evaluation of the model. We also experiment with including eyebrow movements in our data, which otherwise has only the annotations of hand gestures. Additionally, we also do an experiment to find out if the model is generalizable.

We propose improvements to the existing image schema computation algorithm and we also explain our proposed method to represent image schemas as embedding vectors. This work is a step toward generating gestures with the desired property that the gestures should match/be coherent with the speech (see Section 1.3.1), especially toward generating gestures whose shape matches the speech's semantics. With the image schema being representable as vectors, it also becomes possible to calculate the distance between

them, which is a proxy of their difference. Based on those vectors representation, we measure the distances between different image schemas to find out which image schemas are close/similar to each other.

1.6 The Organization Of This Dissertation

In Chapter 2, we explain the background concepts relevant to our work. In this chapter, we introduce the fundamental concepts of gesture, about eyebrow movements, about the representations of gestures, and about image schema.

In Chapter 3, we explain the prior works relevant to our work. In this chapter, we explain about various gesture generators: the rule-based ones, the pre-neural-network ones which rely on the textual input, the pre-neural-network ones which rely on the acoustic input, the recent ones which rely on the textual input, the recent ones which rely on the acoustic input, and the ones which take both acoustic and textual inputs. Besides that, we also discuss works about gesture style and prediction of gesture shape to depict a concrete object. Finally, we close the chapter by explaining the existing limitations; the problems which have not been solved by the existing works.

In Chapter 4, we explain our corpus, the data it contains, and the annotation it contains. We also extract secondary features by using third party programs. For those secondary features, we explain our extraction process and the features we extract. We also give the statistics of the features we extract.

In Chapter 5, we explain about our work on prediction of gesture timing.

In Chapter 6, we explain our work on improving the existing image schema computation algorithm and our work on the representation of image schemas as embedding vectors.

In Chapter 7, we explain our gesture shape representation scheme. We apply this gesture shape representation scheme to annotate our corpus. We also calculate the statistics from our corpus, including the statistics of the gesture shapes related to the image schema.

Finally, in Chapter 8, we close this dissertation with the conclusion of our work.

Chapter 2

Background

In this chapter, we introduce the key concepts we use in this work. We introduce the basic concepts of gestures, its relationship with speech, its classification, and its phases. We also introduce the eyebrow movements because they have one property which is similar to hand gestures: the speech properties which "drive" the hand gestures also drive the eyebrow movements. We also introduce the existing schemes of gesture encoding. To be able to produce gesture automatically, it is necessary to represent the gesture, which in turn necessitates an encoding scheme. Lastly, we introduce the concept of image schema and its relationship with speech and gesture. Some gestures depict an abstract concept, which cannot possibly have any physical shape. The abstract concept is "translated" into a concrete object, which in turn can be depicted by gesture. This "translation" mechanism is done through the image schema concept.

2.1 Gesture Overview

Kendon (1988) proposes an ordering of body movements for communicative purpose according to its stand-alone expressiveness. Those movements, from the least stand-alone expressive to the most stand-alone expressive are gesticulation, language-like gestures, pantomimes, emblems, and sign languages. The more stand-alone expressive the movement is, the less necessary the accompanying speech is. Gesticulation is meaningless without the accompanying speech while sign languages are languages, which means they can be understood by themselves. Besides that, the more stand-alone expressive it is, the presence of language properties increases and the movements are also more formalized/regulated. For example, sign languages are languages, with their own grammars and vocabularies. McNeill (1992) calls this ordering as "Kendon's continuum".

McNeill (1992) uses the term "gesture" to refer to the gesticulation in the Kendon's continuum. In this sense, McNeill defines gestures as idiosyncratic and spontaneous movement of hands and arms which accompany speech. This definition has several properties. The first property is that it is idiosyncratic instead of formalized/regulated. That means, there is no standard on how a gesture should be. Different persons can have different gestures and the same person can use different gestures while uttering the same sentence. A person does not need a special training to be able to do gestures effectively. The second property is that it is spontaneous instead of being done deliberately. Wei (2006) finds that people who are talking on the phone still do gesture. Iverson and Goldin-Meadow (1998)

find that even people who are blind since being born still perform gestures, even when the listener is also blind. The third property is that they accompany speech, which means that they happen at the same time as speech and have no meaning without the accompanying speech. It should be noted, however, not all definitions of gesture limit themselves to hands and arms movement. Calbris (2011) defines communicative gesture as a visible movement of any body part which someone makes in order to communicate. That means, Calbris's definition allows any body parts, not only the hands and arms. Nevertheless, we use McNeill's definition in our work.

McNeill (1992) states that gestures and speech are closely linked. Gestures only occur during speech. They are also co-expressive, which means that gestures and speech express the same or related meanings; they might convey the same meaning or convey complementary meanings, but they work in tandem. Someone might point to the right while saying "the box is on the right", in which case the pointing gesture conveys the location information which is already conveyed by the speech. It is also possible to say "the box is there" while pointing to the right, in which case the pointing gesture and the speech convey a complementary information. The speech expressed that there is a box, but the location of the box is specified only by the gesture. In this case, it is impossible to understand the complete message without observing the gesture. McNeill also gives an example of gestures being used to mark the start of a new topic (i.e., the speaker is changing the conversation topic).

Gesture is also related to spoken language acquisition in children. Goodwyn et al. (2000) find that gesturing helps the language acquisition in infants. Özçalışkan and Goldin-Meadow (2005) find that children combine gesture and speech before they can convey complex information in speech alone. Özçalışkan and Goldin-Meadow (2005) also find that mixing gesture and speech in children signals that the children will soon be able to construct multi-word sentences.

Relationship between gesture and speech can also be observed in aphasia: the impairment of spoken language ability due to brain damage. In Broca's aphasia, the person's ability to produce speech is impaired, but his language comprehension is normal. In Wernicke's aphasia, the language comprehension is impaired. Someone with Wernicke's aphasia can still speak, but his speech is incomprehensible by others. Feyereisen (1987); Sekine and Rose (2013) find people with aphasia are more likely to use gestures in order to compensate for their language inability. Interestingly, Sekine and Rose (2013) find that different types of aphasia have different impact on the type of gestures being produced: people with Broca's aphasia are more likely to use iconic gestures while people with Wernicke's aphasia are more likely to use metaphoric gestures.

Gesture and speech are also synchronous in terms of timing. The most prominent and meaningful part of the gesture, which is called the stroke phase, often occur at the same time as when the important words are being uttered (Loehr (2012)). These are all possible because both gestures and speech are generated from a common process (McNeill (1992)). This relationship is important for this line of research because it opens the possibility to infer the gestures from speech. It should be appreciated, however, the fact that gestures may convey complementary information from the speech means that it is not always possible to infer the gestures from the speech alone.

2.2 Gesture Types

McNeill (1992) splits gestures into four types, namely iconic, metaphoric, deictic, iconic, and beat. This classification is based on the information conveyed by the gesture.

1. Iconic gesture is a gesture where the hand(s) is/are depicting a concrete object or an action. An example of iconic gesture is in Figure 2.2. The accompanying speech of the gesture is "pressing buttons on elevators". In this example, the hand movement depicts the action of pressing an elevator button.

2. Metaphoric gesture is a gesture which depicts an abstract concept. An abstract concept does not have a shape; therefore, the depicted shape comes from a concrete object or action which the abstract idea is transformed into through a metaphor process. An example of metaphorical gesture is in Figure 2.3. The accompanying speech of that gesture is "that allows good ideas to rise to the surface". In that picture, the hands are doing an upward movement as a metaphor for ideas rising up. Idea is an abstract object; therefore, it cannot move in a physical space. However, through metaphor, idea is imagined as a concrete object which can move upward. Despite of the names, the relationship between metaphoric gestures and metaphoric concepts is not bijective. Saund et al. (2019) find that one metaphoric gesture can convey multiple metaphoric concepts. They also find that a sequence of metaphoric gestures can describe a metaphoric scene which is decomposed into multiple metaphoric concepts.

3. Deictic gesture is a gesture which points to an object or location. An example of deictic gesture is in Figure 2.1. The person in the poster is pointing at the reader and the word "you" is highlighted. In this example, both the word "you" and the deictic gesture refers to the same object: the reader of the poster. It should be noted that the object or location being pointed does not have to be in the vicinity of the speaker. For example, a gesture of a hand moving to the right side while accompanying an utterance "there is a bedroom on the right" can also be a deictic gesture even if the bedroom he is talking about is not physically on his right side. Instead, he is imagining that he is inside the apartment, and the bedroom he is talking about is on his right side.

4. Beat is a gesture which is performed by quick back and forth movements along with the speech rhythm. It does not carry a meaning by itself. However, beat gestures mark the discourse context, such as when the speaker is moving to a new topic. An example of beat gesture is in Figure 2.4. He is saying "yes we can to justice and equality" while his right hand is moving up and down. His right hand does not symbolize any meaning, but the movement follows the rhythm of his speech.

Biancardi et al. (2017) call these semantic gestures (i.e. communicative gestures other than beat) as "ideational gesture". Although these gestures carry a meaning, they are not meant to be unambiguous. These gestures cannot be interpreted without knowing what the speaker is saying. It is useful to remember the Kendon's continuum (Kendon (1988)), where body movements for communicative purpose are ordered from the least standalone expressive to the most standalone expressive: gesticulation, language-like gestures, pantomimes, emblems, and sign languages. Ideational gestures are part of the gesticulation in the Kendon's continuum. That means, ideational gestures cannot stand alone, they

Figure 2.1 – An example of a deictic gesture pointing at the reader [a]

[a] https://en.wikipedia.org/wiki/File:J._M._Flagg,_I_Want_You_for_U.S._Army_poster_(1917).jpg

Figure 2.2 – An example of an iconic gesture depicting the action of pressing a button [a]. In this scene, he is saying “pressing buttons on elevators”.

[a] https://www.youtube.com/watch?v=aMcKi1TS2Zs at time 02:01

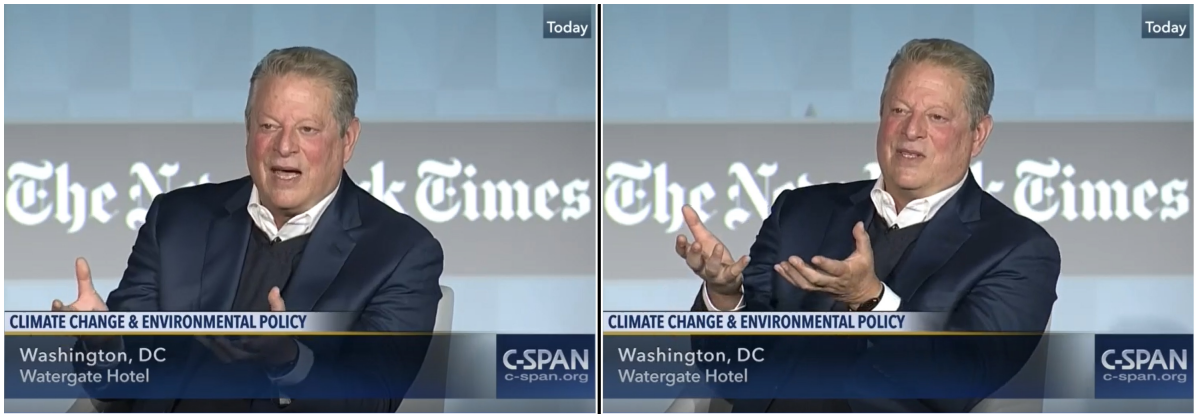

Figure 2.3 – An example of a metaphoric gesture showing an upward movement [a]. In this scene, he is saying “that allows good ideas to rise to the surface”.

[a] https://www.c-span.org/video/?419212-3/vice-president-al-gore-speaks-new-york-times-global-leaders-collecti at time 36:21

Figure 2.4 – An example of a beat gesture showing up and down movements [a]. In this scene, he is saying "yes we can to justice and equality" while his right hand (and also his head) is moving up and down a few times.

[a] https://www.youtube.com/watch?v=Fe751kMBwms at time 11:36

have to be accompanied by speech. On the other hand, emblem and sign languages are unambiguous; they do not need an accompanying speech.

The fact that gesture is not meant to be standalone also means that deciding the type of the gesture requires the knowledge of the accompanying speech and the segment which the gesture coincides. For example, a gesture of an index finger pointing upward is a deictic gesture if the accompanying speech is "the ceiling is dirty" and the gesture coincides with the word "ceiling" because the gesture points to the location of the ceiling, which is above the speaker. On the other hand, the same gesture will be an iconic gesture if the accompanying speech is "you can see a tower" and the gesture coincides with the word "tower" because the gesture depicts the shape a tower, which is long and vertical. This phenomenon that one gesture can have multiple meanings is called polysemy of gestures.

2.3 Gesture Phases

According to Kendon (1980), gestures are characterized by temporal phases, namely preparation, pre-stroke-hold, stroke, post-stroke-hold, and retraction. The stroke phase is mandatory while the other phases are optional.

1. Preparation is a phase where the hand moves from the rest position to the starting location of the stroke.

2. Pre-stroke hold is a phase following the preparation where the hand is still.

3. Stroke is the phase which carries the meaning. This phase takes the greatest effort, is relatively fast, and is well-articulated. Its timing coincides with the important part of the speech, its shape is related to the meaning conveyed in the important part of the speech, and it is performed at a prominent location.

4. Post-stroke hold is a phase following the stroke where the hand is still.

5. Retraction is a phase where the hand is brought to the rest position.

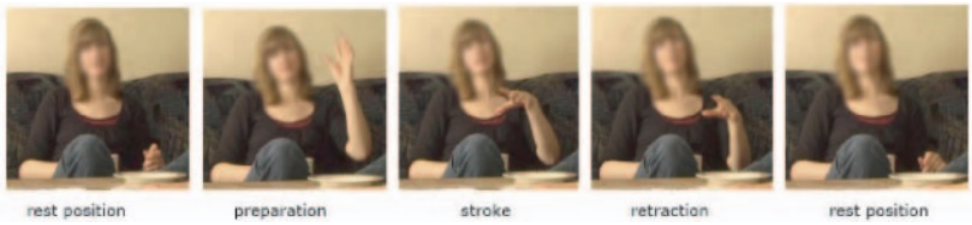

Figure 2.5 – An example of gesture phases from Bressem and Ladewig (2011)

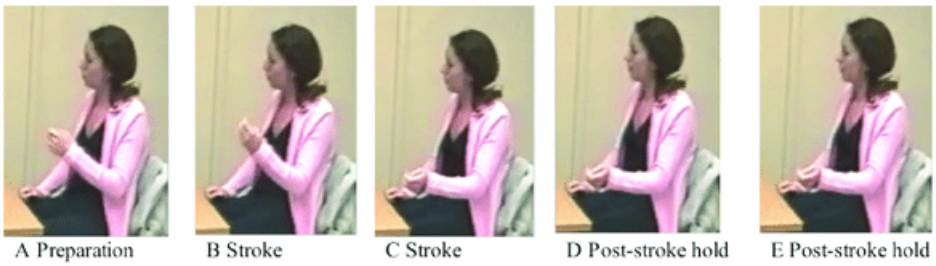

Figure 2.6 – An example of gesture phases from Graziano and Gullberg (2018)

An example of the phases can be seen in Figure 2.5. The gesture starts from the rest position, which in this case the hands rest on the person's lap. Then in the preparation phase, the left hand goes up to the head level. In this preparation phase, the left hand moves to a prominent position: a position where the hand is very visible. Then the person performs the stroke phase, which in this case is a downward movement. The stroke phase is followed by a retraction phase, where the left hand returns to the rest position, which in this case is on the person's lap.

Another example of the phases can be seen in Figure 2.6. In the preparation phase (part A), the left hand moves to the position where the stroke will be performed. The preparation phase is then followed by the stroke phase (part B and C) where the hand moves down. The stroke phase is then followed by a post-stroke hold (part D and E), where the hand is still at the position where the stroke ends.

Not all gestures are "standalone". Successive gestures may co-articulate one from another. That is, when multiple gestures are performed consecutively, the gesture phases can be chained together. On the other hand, beat gestures do not have the same phases. They are often produced with a back and forth movement (e.g. up-down or inward-outward) and mark the speech rhythm.

2.4 Eyebrow Movements

Krahmer and Swerts (2007) state that beat gestures can also be performed by facial and head movements. Specifically, they note that eyebrow movements can be related to beat gestures. Just like a beat gesture creates a perception of emphasis, an eyebrow movement or a head nod also has a similar effect. Similarly, just like a beat gesture creates a perception that the spoken word is more prominent, a rapid eyebrow movement also has a similar (albeit weaker) effect. In a study of professional Dutch newsreaders, Swerts and Krahmer (2010) observe that eyebrow movements tend to accompany the pitch accent. Yasinnik et al. (2004); Flecha-García (2007); Bolinger (1989); Ekman (1979) also observe

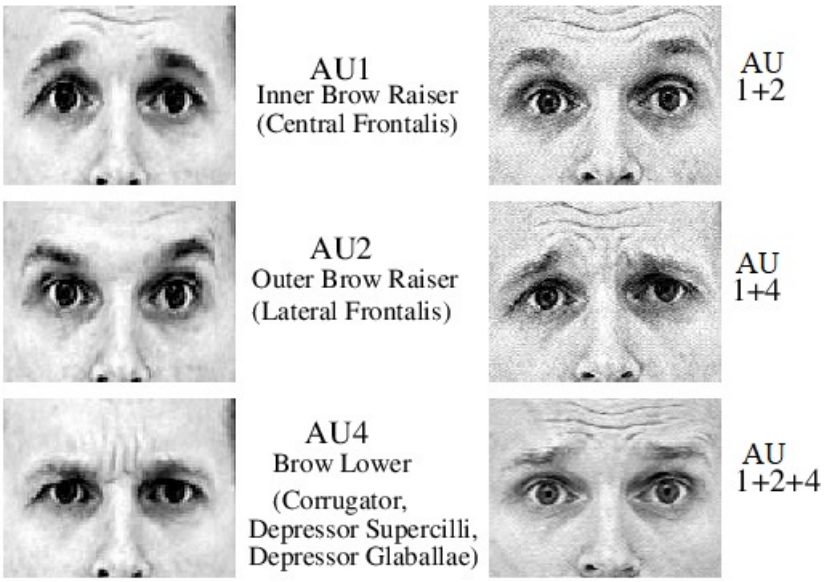

Figure 2.7 – Eyebrow movement Action Units (Bartlett et al. (2002))

that pitch accents are accompanied by eyebrow movements. Eyebrow movements can be encoded by using Facial Action Coding System (FACS) (Friesen and Ekman (1978)). FACS divides a facial movement into the constituent movements. Each constituent movement is called Action Unit (AU). There are three action units (AU) which represent eyebrow movements, namely AU1 (inner brow raiser), AU2 (outer brow raiser), and AU4 (brow lowerer) (Figure 2.7). The presence of either AU1 or AU2 represents rising eyebrow while the presence of AU4 represents lowering eyebrow. These eyebrow movements can occur together as well. In Figure 2.7, we also provide the examples of AU1-AU2 together, AU1-AU4 together, and AU1-AU2-AU4 together.

2.5 Gesture Encoding

There is no standard way to describe gesture shapes. However, several works discuss the important features of gestures which can be used to encode gestures.

Efron (1941) consider the following to be the relevant spatio-temporal parameters of gesture:

- The distance of the hand movement.
- The movement trajectory: sinuous, elliptical, angular, or straight.
- The movement direction.
- The body parts which are involved in the gesture. They can be one of the hands, both hands, fingers, head. If the gesture is done with both hands, a remark on whether they are doing a single gesture.
- The smoothness of the gesture.

Later, Lücking et al. (2016) and Kipp et al. (2007) also propose gesture encoding schemes. Although these two are different works by different authors, their encoding schemes are similar. Their encoding schemes are as following:

- The start and end of stroke are the important parts to be encoded.
- The hand(s) which is/are used for the gesture. If the gesture is performed by both hands, then the two hand movements can be either symmetrical where the two hands are depicting a single entity or they can be asymmetrical where each hand depicts different things. These encoding schemes only consider the hands.
- The movement trajectory; According to Lücking et al. (2016), the trajectory consists of the path (e.g. line, arc, zigzag) and the orientation (away from body, up, left, etc). Lücking et al also include "pointing" as a path. Kipp et al. (2007) only classify the movement trajectory into straight or curved. Kipp et al also include the information about the speed-related properties, such as its abruptness, smoothness, and forcefulness.
- The starting and ending locations; Lücking et al. (2016) represent the locations by using McNeill (1992)'s gesture space (Figure 2.10). The space is divided into zones by concentric squares. The inner-most square is the "center-center", the second-inner-most square is "center", the third-inner-most square is "periphery", and the outer-most square is "extreme periphery". Both the "periphery" and "extreme periphery" are further divided into 8 quadrants: upper, upper left, left, lower left, lower, lower right, right, and upper right. Kipp et al. (2007) represent the locations with 7 possible values in the top-down axis, 4 possible values in the front-back axis, 5 possible values in the front-facing orientation, and 4 possible values in the arm swivel (Figures 2.8 and 2.9).
- The hand shape; Both Lücking et al. (2016) and Kipp et al. (2007) use the American Sign Language alphabets to encode the hand shapes. Kipp et al also mention that HamNoSys inventory (Prillwitz et al. (1989)) can also be used to encode hand shape. Lücking et al also include thumb-up as an additional hand shape. In an interesting agreement, both of them find in their respective experiments that most of the time only a few hand shapes are actually used.

Calbris (2011) proposes a different encoding schema. Calbris describes the gesture with the following parameters:

- The body part(s) which do/does the gesture. These can be the hand(s) or the finger(s).
- The shape of the hands or the fingers while doing the gesture. Because this encoding scheme recognizes both hand and finger gesture, it also recognizes both hand shape and finger shape. Unlike Lücking et al or by Kipp et al who use the American Sign Language alphabets to encode the hand shape, Calbris proposes her own list of the shapes (Figure 2.11).
- The direction where the gesture is pointing. If the gesture is a flat-palm hand gesture, then the direction is where the palm is pointing. In other cases, the direction is where the hand or the finger is pointing.
- The direction of the hand or finger when the gesture is not a flat-palm gesture.
- The movement direction (e.g. upward, downward, etc.).
- The information about the handedness or "fingerness": left only, right only, or both.

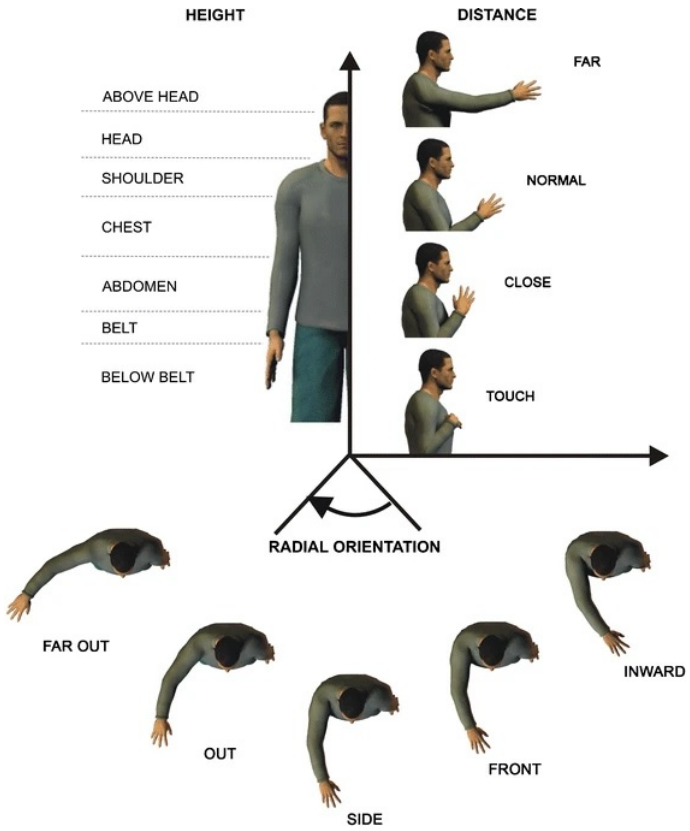

Figure 2.8 – Kipp et al. (2007)'s annotation scheme of the position (top and side view)

Figure 2.9 – Kipp et al. (2007)'s annotation scheme of the position (front view)

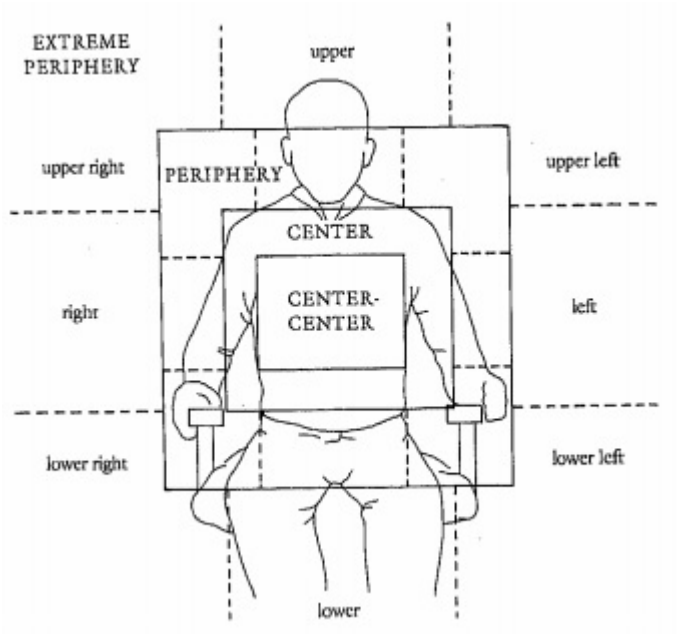

Figure 2.10 – McNeill (1992)'s gesture space

- The region where the the gesture is performed.

We can see that those encoding schemes, from Efron (1941), Lücking et al. (2016), Kipp et al. (2007), and Calbris (2011) have some agreements. All of them take into account the body parts which do the gesture. The movement trajectory is also taken into account in all those encodings. The movement direction is taken into account either directly or indirectly. In the encoding scheme of Lücking et al. (2016) and Kipp et al. (2007), it can be inferred from the starting and the ending locations. However, in the encoding schemes of Lücking et al. (2016) and Kipp et al. (2007), if the movement is so small such that the hand does not move to another region, then the movement direction information will also be lost.

However, we can also observe the difference. Both Efron (1941) and Kipp et al. (2007) include the information about the smoothness or abruptness while Lücking et al. (2016) and Calbris (2011) do not. Both Lücking et al. (2016) and Kipp et al. (2007) take only into account the hands while both Efron (1941) and Calbris (2011) also take into account the finger movements. Efron (1941) does not have the concept of hand shape while Lücking et al. (2016), Kipp et al. (2007), and Calbris (2011) do. Because Calbris (2011) also take into account the finger movement, the shape also includes "finger shape". Calbris (2011) is the only one which takes into account the direction where the hand is pointing. Kipp et al. (2007) has a "weaker" version of this: their annotation indicates where the wrist joint is pointing (see the radial orientations in Figure 2.8 from which the horizontal direction can be inferred and Figure 2.9 from which the vertical direction can be inferred).

2.6 Image Schema

Johnson (2013) defines image schema as a recurrent pattern of reasoning where one entity is mapped into another. For example, "culture" can be mapped into "container" by thinking that some people belong to the same culture while some other people do not.

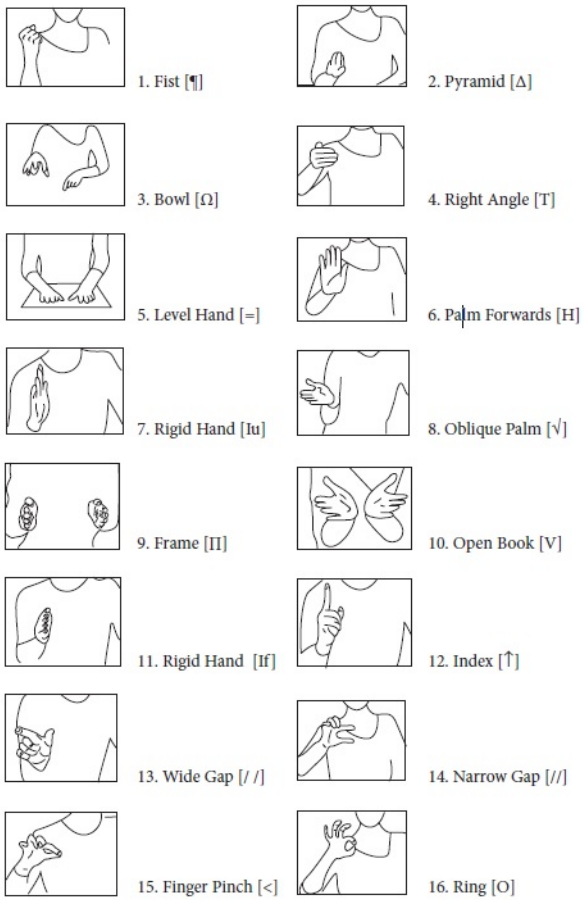

Figure 2.11 – Calbris (2011)'s annotation scheme of the hand/finger shape

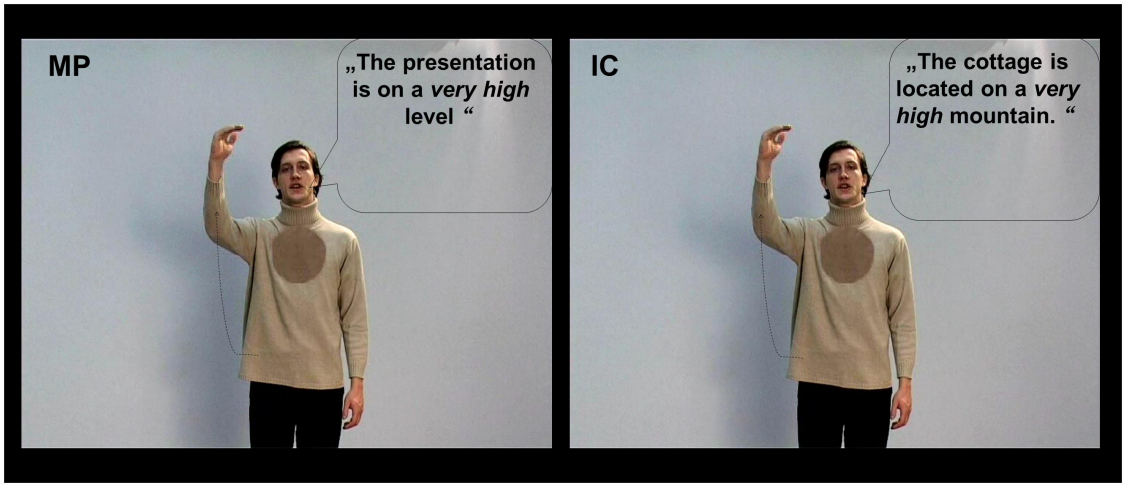

Figure 2.12 – He et al. (2018)'s example of metaphoric gesture (the left part, labelled with “MP”). It can be seen here that the sophistication of the presentation is thought as a concrete object which has the height property (e.g. a mountain), as shown in the right part (labelled with “IC”)

Therefore, culture shares the same property with a container that both of them have a boundary.

Image schema is similar to the concept of conceptual metaphor from Lakoff and Johnson (1980) where human talks about one thing by using another object which has similar properties. For example, in a metaphor “love is a journey”, “love” is imagined to consist of the starting point, the destination, and the path which links both the starting point and the destination. This phenomenon is also observed in our language. In English, we can say “big idea” to mean an idea which has the potential to make a significant impact. However, “big” itself is a property normally used for a concrete object. “Idea” is an abstract object, and therefore it is neither big nor small. Therefore, “idea” has to be mentally mapped into a concrete object which has a physical size.

This “metaphorizing” is relevant because the conceptualization hypothesis states that the way human represents the world in their mind is constrained by the human's physical body Wilson and Golonka (2013), which means there is a need to map an abstract entity into a concrete entity. Metaphor can even affect the physical body movement unconsciously. Miles et al. (2010) find in their experiment that their participants lean forward while thinking about future events. On the other hand, they lean backward while thinking about past events.

Cienki (2013) suggests that this mapping mechanism is how human produces metaphoric gesture. He et al. (2018) show an example where a person rises his hand to describe the high level (of the quality/sophistication) of a presentation (see Figure 2.12). In this case, through “metaphorizing”, the quality/sophistication of the presentation is thought as something which has the height property (e.g. a mountain).

2.6.1 Image Schema and Gesture

Lücking et al. (2016) do an experiment where they ask the participants to perform gestures to manifest various terms, including image-schema terms, by using arm and hand

movements. The purpose of this experiment is to find out if the participants perform similar gestures for the same stimulus.

The image-schema terms they use are taken from Mehler et al. (2015). Mehler et al propose the following image schemas: “container”, “part-whole”, “link”, “center-periphery”, “source-path-goal”, “front-back”, “up-down”, “left-right”, “contact”, “texture”, “near-far”. Lücking et al also add other terms for their experiment. In total, there are 27 terms used in the experiment: “matching”, “zoom-in”, “zoom-out”, “contains”, “whole”, “part”, “back”, “center”, “collection”, “front”, “link”, “periphery”, “down”, “left”, “right”, “rotation”, “up”, “source-path-goal”, “contact”, “far”, “near”, “texture”, “blockage”, “attraction”, “bad/dislike”, “equilibrium”, and “good/like”. It can be observed that each of the “part-whole”, “center-periphery”, “front-back”, “up-down”, “left-right”, and “near-far” image schemas is split into two stimulus terms.

Each participant is asked to manifest 5 or 6 stimulus terms. Therefore, each stimulus term is manifested by 10 participants. Lücking et al find that the participants perform similar gestures for some stimulus terms: “good”, “like”, “bad”, “dislike”, “equilibrium”, “source-path-goal”, “left” and “right”. The stimulus terms of “good” and “like” are often manifested by a thumb-up gesture. The stimulus terms of “bad” and “dislike” are often manifested by a thumb-down gesture. These thumb-up and thumb-down gesture results are expected because of their emblematic nature. For the stimulus term “equilibrium”, the participants often manifest it by stretching out both their arms horizontally and swinging them slightly, which imitates the motion of tightrope walking. For the stimulus term “source-path-goal”, the manifested gesture often imitates the motion of moving an invisible object through the space. For the stimulus term “left”, it is usually performed by the left hand only. On the other hand, the stimulus term “right” is usually performed by the right hand only.

2.6.2 Image Schema Concordance in Different Modalities

In this section, we describe Cienki (2008)'s experiments to investigate the concordance of image schema in different modalities. The idea is to have the same “message” but in different modalities, and then the participants are asked to label those “messages” with the image schema. These image schema labels are then investigated for concordance between the different modalities. Cienki (2008) does three experiments where he asks the participants to label each stimulus with an image schema. The image schemas used in the experiment are “container”, “cycle”, “force”, “object”, “surface”, and “path”. The participant can also choose “other” if he or she feels that none of those predefined image schemas suit the sample. Cienki creates his stimulus based on a set of video-recorded conversation.

The videos are first classified according to their gesture type. Here, the gesture types are “concrete referential” gesture, “abstract referential gesture”, or “other” gesture. Both the “concrete referential” and the “abstract referential” gesture refers to an object, either through its properties, its actions, or its location. The difference between the “concrete referential” gesture and “abstract referential” gesture is on whether the gesture refers to a concrete object or to an abstract idea. The “other” gesture does any other things, such as representing an action (e.g. presenting or dismissing an idea) or discourse structuring (e.g. counting, marking emphasis, etc.). The details about the gesture types are provided in Table 2.1. However, most of the “concrete referential” gestures are pointing gestures, so

Concrete Reference	Abstract Reference	Other
Objects (e.g. a picture frame)	Entities (e.g. the framework of a theory)	Actions (e.g. dismissing, requesting, swearing, hand clapping)
Properties (e.g. the straight edge of a ruler)	Properties (e.g. honesty as straight and solid)	Emphasis (e.g. through beats)
Behaviors and Actions (e.g. the rolling of a tire)	Behaviors and Actions (e.g. the "rolling" development of a process	Structuring (e.g. with counting gestures)
Relative location (e.g. the space behind oneself)	Relative location and relative time (e.g. the past as behind oneself)	Presenting (an idea or argument)

Table 2.1 – The gesture types in Cienki (2008)

the videos with "concrete referential" gestures are excluded, leaving only the videos with "abstract referential" or "other" gestures.

Each video gets its modalities modified, such that each video yields four stimuli with different conditions:

1. With the video but neither the sound nor the transcript
2. With the video, the sound, and the transcript
3. With the sound and the transcript, but no video
4. With the transcript, but neither the video nor the sound

Each participant only labels one condition. There are 20 participants for each condition. Cienki then measures the agreement of the image schema labels between the different conditions. Because the different conditions are actually different set of modalities of the same set of videos, the image schema labelling agreement can be seen as the consistency of the likely image schema across different modalities.

In the first experiment, for each stimulus, Cienki counts the number of participants who label the stimulus with the most frequently chosen image schema for each particular stimulus, and then the distribution is calculated for each condition. For each condition, the choices show higher likelihood than chance and there is a reliable agreement between the four conditions. However, it should be noted that in this experiment, the most frequently chosen gesture is for each stimulus. It does not guarantee that each of the four stimuli which come from the same video have the same most-frequently-chosen image schema. Indeed, on this, the agreement exists only between conditions 1 and 2 and between conditions 3 and 4. For the agreement between conditions 1 and 2, the results are M=9.5 SD=3.23 and M=9.3 SD=2.69 respectively. For the agreement between conditions 3 and 4, the results are M=8.6 SD=2.49 for condition 3 and M=8.7 SD=2.47 for condition 4. When the stimuli are split into those which are associated with "abstract referential" gestures and those which are associated with "other" gestures, there is a greater agreement among those with "abstract referential" gestures than among those with "other" gestures

in conditions 1 and 2. In condition 1, the results are M=10.5 SD=3.28 and for the "abstract referential" gestures and M=8.5 SD=2.91 for the "other" gestures. In condition 2, the results are M=10 SD=2.74 for the "abstract referential" gestures and M=8.7 SD=2.56 for the "other" gestures.

In the second experiment, Cienki counts the number of image schemas chosen at least once for each stimulus, and then the distribution is calculated for each condition and for each gesture type (i.e. "abstract referential" or "other" gesture). The idea is that a lower number of image schemas chosen at least once signifies a greater agreement among the participants. Between the two gesture types, there is only a significant difference in condition 2, where there is a higher agreement for the stimuli associated with "abstract referential" gestures. The results are M=5.0 SD=1.05 for "abstract referential" gestures and M=5.7 SD=0.81 for "other" gestures.

In the third experiment, Cienki counts the number of stimuli in which all six image schemas and the "other" label are chosen at least once for each condition and for each gesture type (i.e. "abstract referential" or "other" gesture). A lower number suggests a higher agreement among the participants. In conditions 1 and 3, there is a higher agreement for the "abstract referential" gestures. In condition 1, the results are 1 stimulus for "abstract referential" gestures and 7 stimuli for "other" gestures. In condition 3, the results are 3 stimuli for "abstract referential" gestures and 10 stimuli for "other" gestures.

Cienki concludes that gestures provide an easily accessible manifestations of image schemas. The "easy" qualifier refers to the fact that gesture is observable by human eyes compared to abstract thinking process in human's brain. Cienki also concludes that gestures can depict/invoke different image schemas from the speech. This is related to our explanation in Section 1.3.1 that speech and gesture can convey complementary information.

2.7 Conclusion

In this chapter, we present the background on the gesture generation problem. We discuss different definitions of communicative gestures, types of gestures, and phases of gestures. We also discuss about the relationship between gestures and speech. The relationship between gestures and speech is the underlying reason which makes gesture prediction from speech becomes plausible. Other than gestures, we also discuss about eyebrow movements. We discuss about the representation of eyebrow movements and that eyebrow movements are related to speech. It can therefore be seen that both eyebrow movements and hand gestures are related to speech. We also discuss about how gestures can be represented. We discuss the principal works on the gesture representations and both their similarities and differences. Lastly, we discuss about reasoning pattern called "image schema". We give a brief introduction about image schema and we discuss about its relationship between image schema and metaphoric gesture, including the relevant experimental findings. We also explain the experimental findings about the concordance of image schema in different modalities.

Chapter 3

Related Work

Existing gesture generators work by taking in the text or the acoustic as the input and generate the gestures accordingly. It means that the systems assume that gestures can be inferred according to the corresponding text or acoustic. However, according to McNeill (1992), gestures and speech are generated from a common process. McNeill also states that in some cases gestures and speech complement each other instead of conveying the same information. In this case, it is impossible to infer the gestures according to the speech. It breaks the underlying assumption of existing gesture generators. Yet, by simplifying the relationship between speech and gestures into that gestures can be computed according to the speech, the problem of gesture generation becomes tractable because speech can be easily observed, unlike the thinking process in human's brain. The earliest gesture generators are rule based systems; i.e. they generate gestures according to the set of rules gathered from the literatures. However, because of the complexity of the rules, machine-learning based systems are developed so that the rules can be learned automatically.

3.1 Rule-Based System

The earliest gesture generators are rule based systems; i.e. they generate gestures according to statically defined set of rules. The rules come from literatures or a prior statistical analysis. BEAT (Behavior Expression Animation Toolkit) by Cassell et al. (2004) is one of the pioneers of these systems. BEAT takes the text as its input and then generates the gestures accordingly. BEAT processes the text by parsing it and then marking where the new topic starts. The identification of where the topic starts is done because Cassell et al consider gesture to be less likely to happen when the topic just started. The parsing is done so that verb phrase and noun phrase can be distinguished from each other. The differentiation between verb phrase and noun phrase is important because BEAT has different sets of rules for actions (i.e. verb phrase) and object (i.e. noun phrase). There is also a word tracker to decide if a word is newly encountered or has been seen earlier, because Cassell et al consider that there should be some gestures when there is a new word. Lastly, BEAT also checks for contrast between adjectives by using WordNet (Miller (1995))'s synonym and antonym relationships because Cassell et al consider there should be some gestures at the contrast. The architecture and the processing pipeline of BEAT can be seen in Figure 3.1.

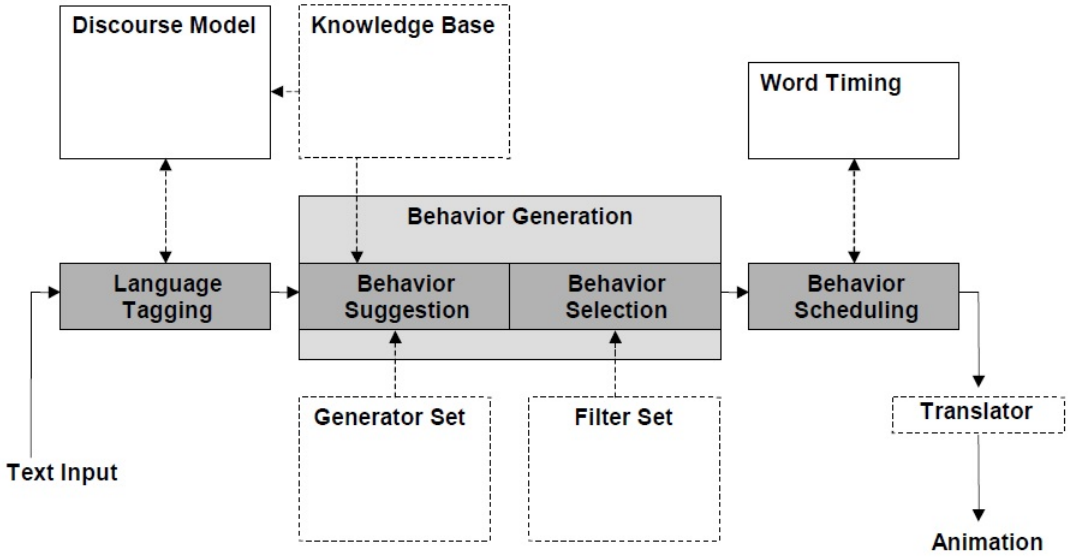

Figure 3.1 – The architecture and the processing pipeline of BEAT (Cassell et al. (2004))

NVBG (Non-Verbal Behavior Generator), by Lee and Marsella (2006), is another gesture generator. Unlike BEAT whose rules come only from literature, NVBG also contains rules extracted from statistical analysis of non-verbal behaviors observed in conversation videos. Unlike BEAT, NVBG also takes into account the conversation context. The conversation context consists of the "affect", the "coping", the "emphasis", and the "turn". The "affect" is the affective state of the agent (e.g. joy, distress, etc.). The "coping" is the coping strategy employed by the agent. The "emphasis" is the part of the text input which should be emphasized. The "turn" is the turn-taking action which the agent should take (i.e. "take", "give", or "keep"). The idea is that this conversation context affects the non-verbal behavior. For example, McClave (2000) finds that head nods are signs of affirmation or backchannel. Affirmation itself is related to the coping strategy while backchannel is related to the turn taking. In another example, Ekman (1982) finds that eyebrow movements are signs of emphasis. These conversational context parameters are especially relevant because NVBG does not take the acoustic as input. It might be possible to infer some of the conversation context from the acoustic, but the acoustic is not available as an input. The text input is parsed into phrases. These phrases and words, along with the conversational context, are then mapped into pre-defined actions (e.g. interjection, negation, affirmation, etc.) according to the aforementioned rules which are extracted from literature and statistical analysis of conversation videos. Then, there is another mapping from the action to the gesture from which the gesture is generated. The architecture and the processing pipeline can be seen in Figure 3.2.

Cerebella, by Lhommet and Marsella (2013), is another gesture generator. Unlike both BEAT and NVBG, Cerebella takes the acoustic as an input as well, on top of the text input. However, the acoustic input is only limited to the word stress and the overall agitation. The overall agitation is a class which indicates whether the voice is breathy, tense, or modal (i.e. normal). The classification comes from the output of Scherer's classifier (Scherer et al. (2013)). The inputs are then processed to obtain the communicative functions. Similar to BEAT and NVBG, the text is parsed into phrases. There are five computations of communicative function, namely "emphasis analysis", "emotion analysis", "initial

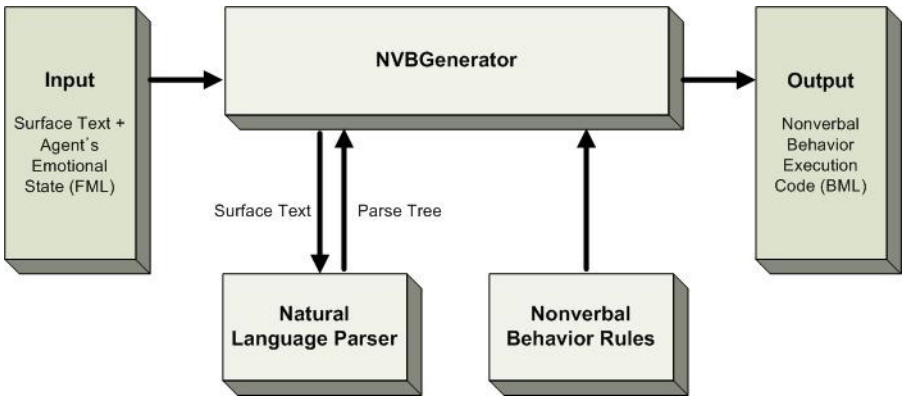

Figure 3.2 – The architecture and the processing pipeline of NVBG (Lee and Marsella (2006))

lexical analysis", "initial rhetorical analysis", and "knowledge elaboration analysis". The "emphasis analysis" infers which words are being emphasized according to the acoustic's word stress input. The "emotion analysis" decides the arousal level (high, medium, or low) based on the acoustic's overall agitation. The "initial lexical analysis" works by mapping phrases/words into a set of pre-defined communicative functions by using WordNet (Miller (1995)) and by classifying whether the noun refers to an abstract concept or a concrete object. The abstract concept or concrete object classification is relevant because metaphoric gestures are often used to represent abstract concepts. The "initial rhetorical analysis" infers the rhetoric-related communicative functions (limited to comparison and contrast only). The "knowledge elaboration analysis" is to combine the previously-computed communicative functions. A part of its process is to resolve conflicts between different communicative functions and semantic disambiguation. Each of the computed communicative functions is mapped to a set of possible gestures. Having multiple possible gestures for each communicative function allows the agent to perform more varieties of gestures. The architecture and the processing pipeline can be seen in Figure 3.3.

The rule based systems have an inherent problem of the rule complexity. The rules governing the relationship between speech and gestures are complex and are still being studied. This problem led to the development of machine-learning based systems where the relationship is automatically learned instead of being manually set by its creators. Traditionally, the machine-learning based systems are categorized into either a semantic-based system or an acoustic-based system. A semantic-based system computes the gesture output according to the text input. An acoustic-based system computes the gesture output according to the speech acoustic. However, this division also affects the type of the generated gestures because beat gestures are related to the speech acoustic while ideational gestures are related to the semantics. That said, there is a recent work which tries to use both the semantic and the acoustic. There are also recent works which address new problems which were not addressed in rule-based systems.

3.2 Pre-Neural-Network Semantics-Based System

These systems extract semantic features from the text and then learn the relationship between those features with the gestures. Because they do not use the acoustic information, naturally they do not generate beat gestures. Instead, they generate ideational gestures.

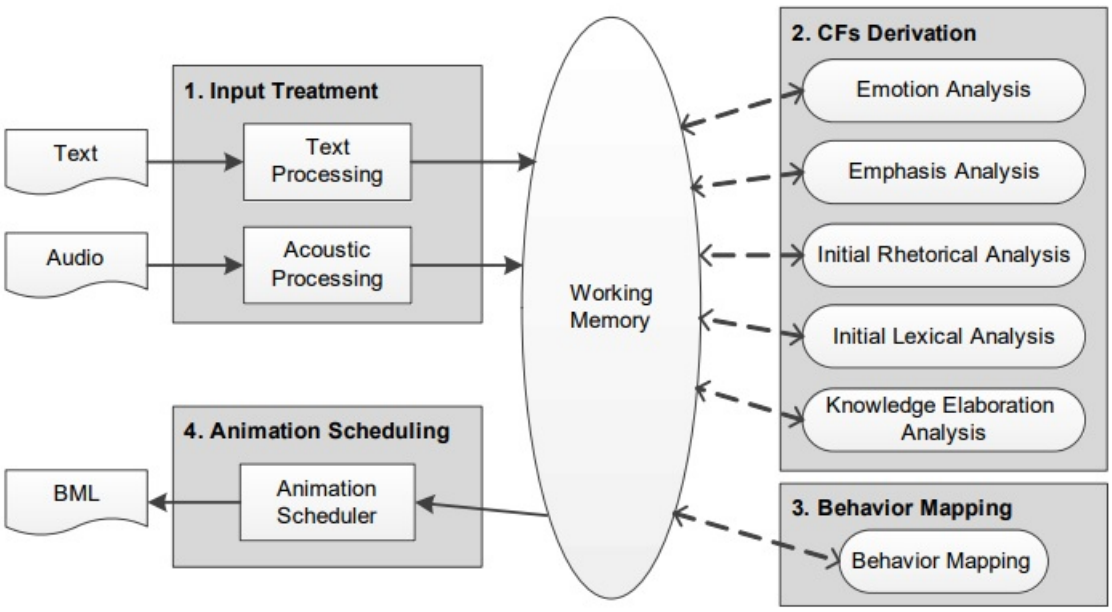

Figure 3.3 – The architecture and the processing pipeline of Cerebella (Lhommet and Marsella (2013))

Bergmann and Kopp (2009)'s technique generates iconic gestures by using Bayesian decision network. The learned data comes from their experiment where they ask dyads to converse. The conversation topics are giving directions and describing landmarks. The system computes the gestures based on the referent features, the discourse context, and the previous gesture. The reference features themselves are whether the gesture can be decomposed, its symmetry (if exists), its main axis, its 3D position, and its shape type. The discourse context features are the structure (introduction of the topic or the content of the topic), whether it is private or shared, and the communicative goal. The gesture information, however, is in the form of a vector of classes. For example, part of the gesture information is the finger orientation whose possible values are up, down, left, right, towards, and away.

Ishii et al. (2018)'s technique is based on Conditional Random Field to generate the body motion based on the natural language analysis features. Those features are the phrase lengths, word positions, bag of words, dialogue acts, parts of speech, and large-scale thesaurus. The features are extracted beforehand. Similar to Bergmann's technique, the motion is represented as a vector of classes. Ishii et al's technique, however, does not model the temporal dependency because there is only one output vector for each sample.

Lhommet and Marsella (2014) propose a technique which generates metaphoric gestures from communicative intention. Communicative intention by itself is obviously abstract and wide ranging, so they limit it into three communicative intentions: depicting a property of an element, adding or removing an element to a set, and contrasting the property of two elements. These communicative intentions are then mapped to concrete objects by using OpenCyc ontology [1], which are then mapped to gestures according to a set of rules. The theory underlying this approach is image schema (Johnson (2013)), which states that human reasons by mapping one entity into another (see Section 2.6).

[1] http://www.cyc.com

Ravenet et al. (2018a,b) develop a technique which computes the shape of metaphoric gestures according to a representation of the text. This is done by computing the image schema from the text. A set of rules then decide the gesture shape according to the image schema. Ravenet et al's technique works by firstly parsing the sentence to get the lemmas and their parts of speech. Then, by using a simplified Lesk algorithm, it decides the actual sense of the lemma. From the actual sense, then it traverses the WordNet sense graph through the hypernym edges until it finds a sense which has been specified to belong to a certain image schema. There are two rules in this rule-based technique, firstly the list of image schemas and their corresponding WordNet senses, and secondly the gesture shape for each image schema.

Ravenet et al's technique is innovative because it provides a way to compute the image schema from the text. Unlike the technique of Lhommet and Marsella (2014), the technique of Ravenet et al takes a free-form text. The likely existence of relationship between image schema and metaphoric gesture has been discussed in literature (see Section 2.6), but Ravenet et al's technique is able to automatically infer the image schema from a free-form text input.

3.3 Pre-Neural-Network Acoustics-Based System

These models learn the body movement according to the speech acoustic. Because they do not use the semantic information, they only generate beat gestures. A common feature among these techniques is that they express the problem as a sequence prediction problem: the input is a sequence of acoustic features while the output is a sequence of movement.

Levine et al. (2010), the pioneer of this approach, use Conditional Random Field to model the sequential dependency. Levine et al use fundamental frequency, intensity, and the lengths of each syllable as their input. This technique requires a motion library. However, the technique has a constraint such that the chosen consecutive motion segments from the library must connect smoothly with each other. This constraint was resolved in Chiu and Marsella (2014).

Chiu and Marsella (2014) use normalized amplitude quotient, peak slope, fundamental frequency, energy, energy slope, spectral stationarity, and the voice tenseness as their input. They add Gaussian Process Latent Variable Models to ensure the movement's smoothness. This addition enables the technique to generate smooth transition between positions, and thus obviating the need of an extensive motion library to generate smooth motions, and thus solving the problem at Levine et al. (2010). However, the technique of Chiu and Marsella learns the mapping from prosody to motion in two steps, namely mapping from prosody to discrete gesture annotations and mapping from the discrete gesture annotations to the motion. Thus, the discrete gesture annotation is an information bottleneck.

Bozkurt et al. (2016) use Hidden Semi-Markov Model to model the sequential dependency with intensity, fundamental frequency, and confidence-to-pitch as input. Similar to the technique of Chiu and Marsella, the technique of Bozkurt et al is not limited by a motion library to generate smooth motions because the technique can automatically smoothen the transitions. However, Bozkurt et al's technique internally does clustering of the acoustic input and convert the acoustic input values into the cluster which the inputs belong to. Effectively, it converts the real-valued inputs into a class. The technique does

the same to the gesture. Effectively, it means there are information bottlenecks because the real-valued acoustic and gesture information is compacted into classes.

3.4 Recent Semantics-Based Neural-Net Techniques

Ahuja and Morency (2019) develop a technique which predicts the body movement from the text input. Fundamentally, this is achieved by creating a joint embedding of the text and body movement in the same latent space.

The text itself is processed at the sentence level and is processed by using a word embedding technique. The word embedding technique used is Word2Vec (Mikolov et al. (2013)). Each word is represented as a vector by using Word2Vec, then they are chained together into a sequence. The use of word embedding to represent semantic is innovative because it represents the text as vectors. Vectors are convenient to use in machine-learning based systems. Word-embedding vectors themselves are special because two such vectors, if they are close to each other, then they are expected to have similar meanings (Jurafsky and Martin (2000)). Thus, word embedding allows us to create a notion of how different two words are. Before there is word embedding, we can say if two words are different, but we cannot say whether the two words differ a little or a lot.

Ahuja and Morency (2019) represent the pose as the coordinates of the joints. These joint coordinates are chained together to form a sequence to represent a movement. Therefore, both the text and the body movement are represented as sequences. The method to infer the movement from the text has three components. The first component is that the pose representation is compacted into its latent representation by using an autoencoder. The second component is that the sentence representation is mapped into the vector space of the aforementioned latent representation of the pose. This second component is the core idea of the method: that both the text and the body movement are mapped to the same latent space. The third component is that the distance between those two vectors is minimized, so effectively the representation at the latent vector space becomes the joint embedding of both the text and the pose. Ahuja and Morency do both objective and subjective evaluations to compare their method against Lin et al. (2018). The objective evaluation is done by comparing the average position error, which is basically the average displacement of the joints from the ground truth. The method of Ahuja and Morency performs better on the objective evaluation. For the subjective evaluation, Ahuja and Morency ask the respondents “Which of the 2 generated animations is better described by “<sentence>”?” where one animation is from the output of Lin et al. (2018)and the other one is either from the output of Ahuja and Morency (2019) or from the ground truth. The output from Ahuja and Morency's method is rated more favorable than the output of Lin et al's method, but less favorable than ground truth.

In the gesture generation problem, the use of word embedding to represent the text input opens a new line of techniques to represent the semantics in a way suitable for machine learning. Representation learning itself is still an active research area, so there can be new and better ways to extract the semantics from the text.

3.5 Recent Acoustics-Based Neural-Net Techniques

The recent acoustic-based systems, similar to the pre-neural-network ones, also express the problem as a sequence prediction problem: a sequence of acoustic feature representa-

tion as the input, a sequence of body movement representation as the output. However, unlike the pre-neural-network techniques, the process in the new techniques does not compact the acoustic feature values into a few classes only. Similarly, the movement is also not compacted into only a certain number of possible movements.

Hasegawa et al. (2018) use Bi-Directional Long Short-Term Memory Hasegawa et al. (2018) with Mel-Frequency Cepstral Coefficients as their input. Kucherenko et al. (2019) extend the work of Hasegawa et al by compacting the representation of the motion by using Denoising Autoencoder. They also experiment with other prosodic features, namely the energy of the speech signal, the fundamental frequency contour logarithm, and its derivative. They compare their results against the results of Hasegawa et al. In their objective experiment, they find that their model's results have lower jerkiness than the results of Hasegawa et al's model. Besides that, in their another objective experiment they also find that the results of their model are more similar to the ground truth than the results of Hasegawa et al's model in terms of average acceleration. Besides that, in their subjective experiment, they find that the result of their model is perceived to be more natural than the result of Hasegawa et al's model.

Ginosar et al. (2019) use UNet (Ronneberger et al. (2015)) with the timestep as one of the dimensions to model the sequential dependency and use Mel-Frequency Cepstral Coefficients as their input. They also add an adversarial learning component to so that each possible input can have multiple correct output. They evaluate their model through perceptive study and find that the adversarial learning component makes the resulting agent's animation to be more similar to human's gesturing style. The important contribution of this work is that the system learns a many-to-many relationship from speech into movements. Previously, for each input, there is only one correct output. Besides that, once the learning is done, the system would be deterministic: given the same input, the system would yield the same output. On the other hand, the many-to-many relationship enables a stochastic output. In the gesture generation problem, it enables the agent to say the same thing twice, yet still produces different communicative gestures.

Ferstl et al. (2019) expand the use of adversarial learning further. They use multiple discriminators to evaluate the generated motion according to several qualities: phase structure, motion realism, intra-batch consistency, and displacement. They use fundamental frequency and Mel-Frequency Cepstral Coefficients as their input. Interestingly, embedded within their model architecture, there is a phase classifier. The classifier takes the three-dimensional velocities of 16 joints as inputs. The 16 joints correspond to the joints of the arms, hands, and fingers. From the joints taken as the inputs, we can see that these parameters attempt to capture the hand gesture. The classifier also takes the fundamental frequency as another input. The output of the phase classifiers are the phases. The phases used are based on Kendon (1980)'s gesture phases. Specifically, the phases used by Ferstl et al are "preparation", "pre-hold", "stroke", "hold", "independent hold", "rest hold", "partial retract", "retract", and "none". The purpose of the phase classifier is to enforce of a realistic phase structure. For example, a preparation cannot be immediately followed by a retraction.

It should be noted, however, although adversarial learning enables stochastic output, it does not guarantee that the output will actually be diverse. Wu et al. (2021) develop a technique based on Conditional Generative Adversarial Network to generate upper body movement. They use the fundamental frequency, the intensity, and the first and second derivatives of the fundamental frequency and the intensity. The technique has two variants: with an Unrolled Generative Adversarial Network and without one. The Unrolled

Generative Adversarial Network is a method to increase the diversity of the output. In their objective study, they find that the output distribution when they use the Unrolled Generative Adversarial Network is more similar to the ground truth than when they do not use it. However, in their subjective study where they ask the human participants to rank the generated movements according to their naturalness, the system with the Unrolled Generative Adversarial Network performs similarly than the system without one.

3.6 Semantics+Acoustics Based System

Kucherenko et al. (2020) develop a technique to generate body movements based on both the text and the acoustic of the speech. The text is represented by BERT (Devlin et al. (2018)). Kucherenko et al use the log-power mel-spectogram to represent the acoustic, following Ferstl and McDonnell (2018); Ginosar et al. (2019). The schema of the technique is in Figure 3.4. Kucherenko et al use a moving time window to define one sample. Within each time window, which contains several frames, at each frame the text feature and the audio feature are extracted, concatenated, and fed into a feed-forward neural network layer. This layer acts as an encoder to reduce the dimensionality of the data. The outputs of this layer are then concatenated to create a unified representation of the entire input at that time window. This is then passed into many neural-network layers. The model is also fed with the output of the previous time window to express the time-dependency (i.e. the output of this time window is affected by the output of the previous time window). They perform a subjective study for ablation purpose where, among others, they remove either the audio input or the text input. The human respondents are shown pairs of videos. In each pair, there is a video from the full model and there is another video from the model whose one of the input modalities is removed. The respondents are then asked four questions (Figure 3.5). When one of those input modalities is removed, the respondents mark them worse than the full model in all the four questions. This is not surprising, because when one of the input modalities is removed, then some information (e.g. speech rhythm and the semantics) will no longer be available. However, surprisingly, the outputs of the system without the text input is rated lower than the outputs of the system without the audio input, even at the questions which are about the audio-related properties, namely Q1 (In which video the character's movements most human-like?) and Q4 (In which video are the character's voice and movement are more in sync?). Besides that, in their other subjective study where they compare the performance of their system against the system of Ginosar et al. (2019) which take only acoustic as the input, although the system of Kucherenko et al performs better in both the human-likeness and on reflecting what the character says, it is notable that the performance difference is smaller on "Q2: In which video do the character's movements most reflect what the character says?" (Figure 3.6). This Q2 question is related to the semantics. Because the system of Kucherenko et al takes both the audio and the text as inputs while the system of Ginosar et al only takes the audio as input, the system of Kucherenko et al should perform much better on the semantic quality. However, as can be seen in Figure 3.6 Q2, the preference toward the system of Kucherenko et al against the system of Ginosar et al is only around 0.1 even though the system of Ginosar et al does not even take the text input. As a comparison, in Figure 3.6 Q1, which measures a speech rhythm related quality, the preference toward the system of Kucherenko et al is around 0.25.

These results suggest that making use of both acoustics and text input effectively is not a trivial problem. While using both modalities enable us to learn more information, it

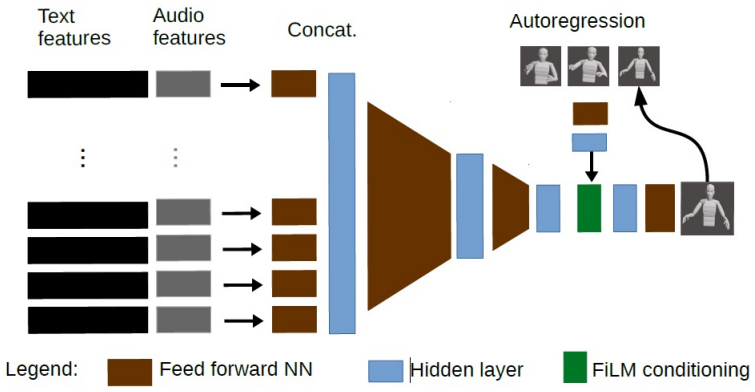

Figure 3.4 – Kucherenko et al. (2020)'s technique

does not guarantee that we will effectively learn more information. For example, it is not certain if BERT is an ideal way to extract and represent the semantics from the text. Even if an ideal embedding to represent the text is known, a more sophisticated technique to combine both the acoustics and the text modalities might still be needed. Effective use of both modalities, namely the acoustics and the text, is still an open problem.

3.7 Gesture Style

There are several works about gesture style. These works learn the gesture style of an individual person and then they produce synthetic gestures with the style of a certain person.

Neff et al. (2008) develop a system which takes a text as input and then generates a gesture in the style of a specific person. The style is learned from data. The learning works by firstly taking a video and its transcript and breaking the speech into phrases. Each phrase is then classified into a semantic class. Meanwhile, the hand movement is broken into its own "phrases" and each of them is classified into a gesture shape class. The style of each person is defined as the probability that he performs a certain gesture shape given the semantic class. For the generation, the system takes a text, breaks it into phrases, gets the corresponding semantic class, and then chooses the corresponding gesture shape class according to the aforementioned probability. In their subjective study, Neff et al run an experiment with two persons being the gesture style template, and therefore there are two possible styles. The experiment consists of two parts. In the first part, the participant watches a video and is asked whose style is being used in the video. For this part, there are roughly the same number of videos of either style. The participants answer correctly 69% of the time, which is above the chance. In the second part, one video for each style is shown, and therefore there are two videos, and the user is asked to map the video to the correct style. The participants answer correctly 88% of the time, which is above the chance as well. These results suggest that Neff et al successfully produced the style.

Ahuja et al. (2020) also develop a system which generates gestures with the style of a specific person. The system is based on a neural network generative model. They cast the gesture generation problem as a style transfer problem. The style transfer technique was initially applied for painting (Gatys et al. (2016)), but afterwards it is also applied at other problems such as videos and speech (Ahuja et al. (2020)). In the style transfer problem, a sample is decomposed into its content and its style. A synthetic sample is then generated

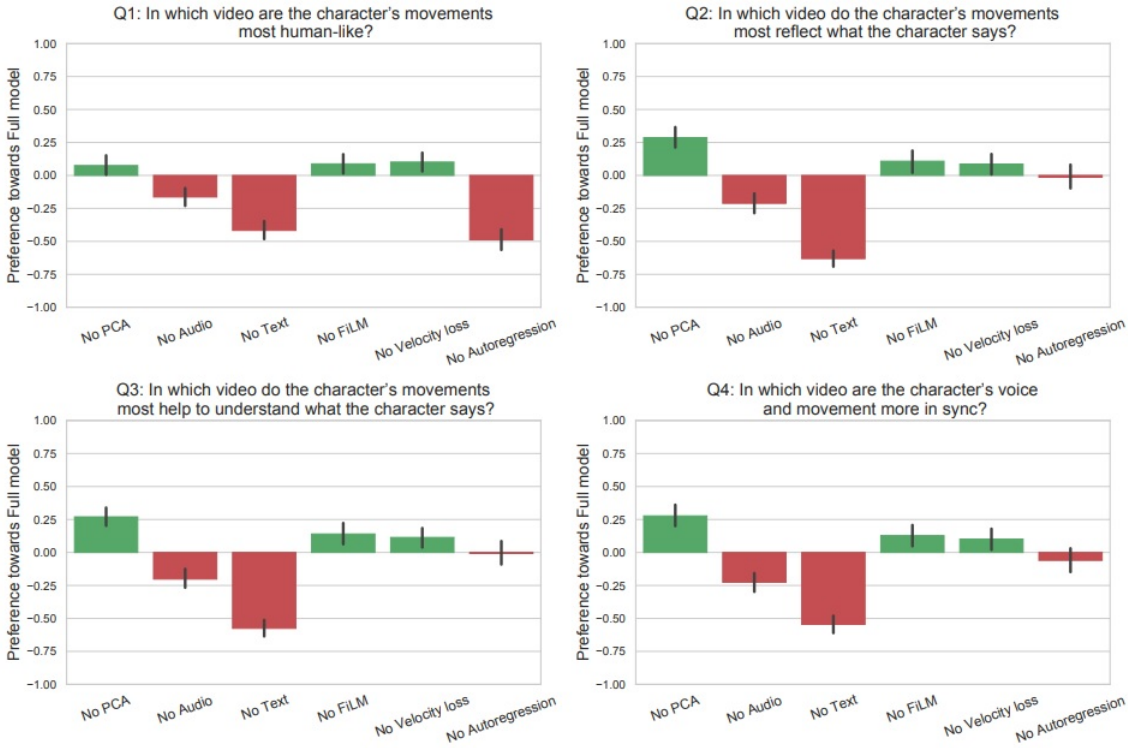

Figure 3.5 – Kucherenko et al. (2020)'s ablation study result

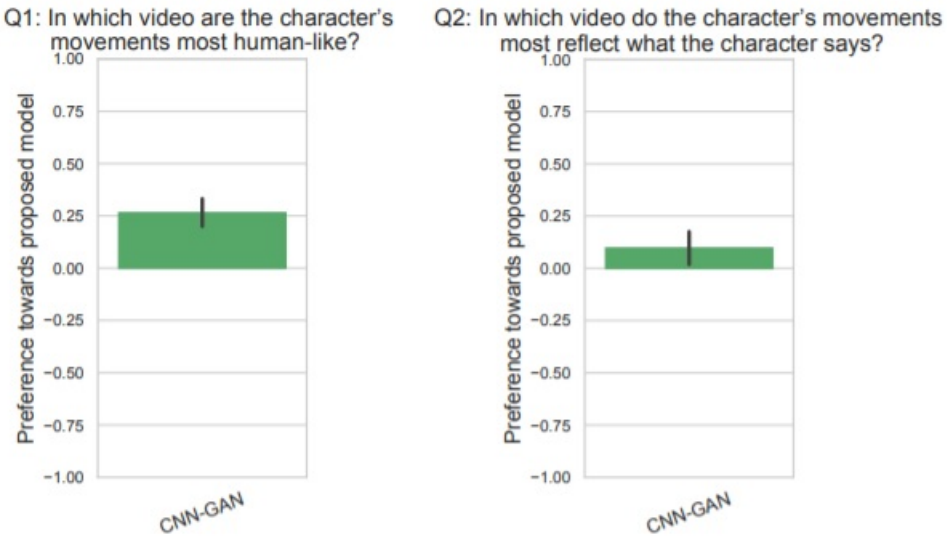

Figure 3.6 – Kucherenko et al. (2020)'s comparison of their technique against the baseline system from Ginosar et al. (2019)

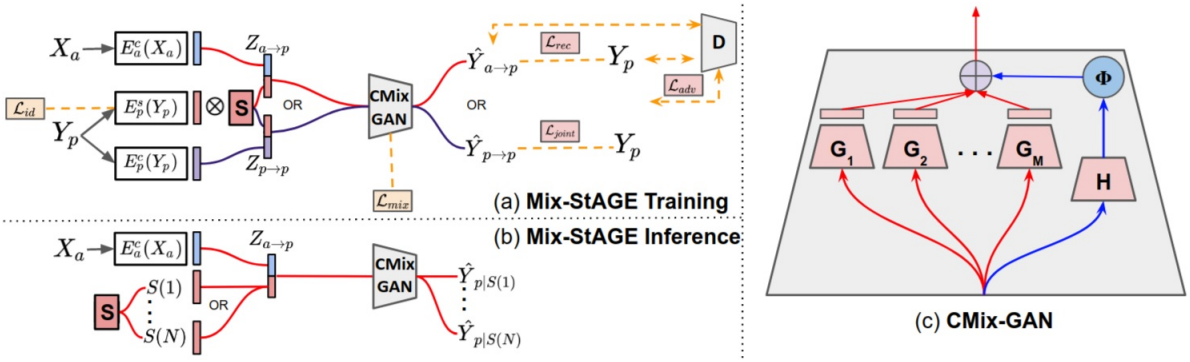

Figure 3.7 – Ahuja et al. (2020)'s technique

with the same content but a different style. The content and the style are learned from the data instead of being explicitly defined. Specifically, the training samples are annotated on which style they have: one sample has one style, but one style has many corresponding samples. In the case of Ahuja et al. (2020), one person is one style. Therefore, effectively, the *de facto* definition of style is the common properties of the different samples which come from the same person. Unlike the system of Neff et al. (2008) which takes the text as the input, the system of Ahuja et al takes the acoustic as the input. Unlike the system of Neff et al, the system of Ahuja et al does not have a formula which explicitly defines a style. The schema of Ahuja et al's technique is in Figure 3.7. In part a of the schema (Mix-StAGE Training), where the learning part is shown, the Y_p pose input is fed into two encoders, E_p^s which will extract the "style", and E_p^c which will extract the "content". The matrix S will contain the learned style vectors. In part b of the schema (Mix-StAGE Inference), X_a is the audio input and the S is the embedding/vector of the chosen style. The value of the vector is obtained from the training. Ahuja et al conduct a subjective study to test the style transfer capability. The human respondents are shown a pair of videos: one video from the ground truth and the other one is generated by their system. The two videos may or may not have the same style. There are four possible styles. The respondent is asked whether the two videos have the same style. However, the percentage of the correct answers is not very high, only around 20% , which is lower than chance. It is possible that their style embedding is not enough to characterize each style. However, this technique is still interesting, because it enables learning the style without creating an explicit formula of what constitutes a style. Besides that, similar techniques have been successfully applied for other problems, so it is possible that the future extensions of this technique will perform better.

3.8 Iconic Gesture Shape

Nihei et al. (2019) use pre-trained VGG-16 neural network (Simonyan and Zisserman (2014)) to statically learn iconic gestures from the images of the objects those gestures represent. VGG-16 is a neural network model for object recognition in an image. This neural network works by having many layers where different layers have different resolutions. The early layers have high resolution with a lot of details of the image retained. However, in the deeper layers, the details are eliminated and only the basic shape is retained. Nihei et al feed images of concrete objects into the pre-trained VGG-16 network, and then extract the underlying basic shapes of the objects from a deep layer of the net-

work. These basic shapes can then be reproduced as iconic gestures. This technique is innovative because although iconic gestures shows the simplified shape of the object, the way to get the simplified shape was not known. However, this technique obtains the basic shapes without any context. Iconic gestures are used for communication, so the shape which is shown has to be relevant to the feature being spoken. This context is not taken into account in the technique.

3.9 Existing Limitations

The existing works have various limitations and unexplored questions. We would like to focus on some of those limitations and questions. The first limitation is that even though there are works which attempt to generate gestures based on either the text, or the acoustic, or both, there has not been a work which attempts to tell when to perform beat gestures and when to perform ideational gestures. This is an important information. When an ideational gesture is performed, it means the gesture is conveying a semantic information. The gestures should depict the relevant semantic information, and therefore the relevant information is from the text. On the other hand, when a beat gesture is performed, it means the gesture is there to match the rhythm of the speech and the discourse context (e.g. emphasis). In this case, the relevant information comes from the acoustic features. This is the motivation why we work on the gesture timing prediction. We want to know when ideational gestures are performed and when beat gestures are performed. It should be noted, however, that the difference between ideational gestures and beat gestures is not only about the form: a hand moving up and down might be beat gestures but might also be an ideational gesture depicting an object moving up and down. The distinction must be made in the context of the accompanying speech. It should also be noted that the categorization is actually not "pure": a gesture can have both beat properties and ideational properties at the same time, but we still can say that a certain gesture is more beat or more ideational (McNeill (1992)). Therefore, the binary classification into beat and ideational in this discussion is a simplification.

The second limitation we want to address is obtaining the ideational gesture shape, especially metaphoric gesture. This gesture has to actually depict the abstract concept being conveyed. The image schema concept which is used by Ravenet et al. (2018a,b) is relevant for this. The image schema is derived from the semantics of the text. Although image schema and its relationship with metaphoric gesture have been discussed in the literature, it is not clear how to adapt them into a machine-learning based method. The difficulty is that machine learning techniques take vectors as inputs, therefore the image schema has to be converted into a vector first. However, it cannot be a random vector. A simple solution is to represent the image schema as a one-hot representation, but one-hot representation ignores the degree of difference between different image schemas. Newer representation techniques, on the other hand, map the nominal objects to vectors in such a way such that if two nominal objects are similar to each other, they will tend to be mapped to two nearby vectors. Word embedding techniques also work according to that principle. Thus, it is interesting to investigate how we can represent image schemas as vectors while taking into account the similarity between them. This requires us to define that the notion of image schema similarity. If we can map image schemas into vectors, then we will be able to use image schema as an input of a machine learning model, which in turn can then be used to learn gestures.

The third limitation is the way the texts are represented by using word embedding in the recent machine-learning based systems. Ahuja and Morency (2019) use Word2Vec while Kucherenko et al. (2020) use BERT to represent the text. Although word embedding works in such a way that the vector representations of two similar words will also be likely close to each other, different word embedding models define the word similarity differently. There are also other word embedding techniques other than Word2Vec and BERT. It is interesting to investigate which word embedding technique is more suitable to represent the text input for a gesture generation problem.

3.10 Conclusion

The earliest works on automatic gesture generation are rule-based systems. The rules are extracted from literature or statistical analysis of some corpus. However, because of the complexity of the rules, machine-learning based systems are developed. These systems automatically extract the rules from the data. Some of these systems take only the acoustic features as the inputs while some other take only the text as the input. The works which use the acoustic features as the input tend to express the problem as a sequence prediction problem where the input is a sequence of acoustic features and the output is a sequence of gesture motion. The recent works which takes the text as the input use word embedding to represent the text input. There is also a recent work which uses both the acoustic features and the text as the inputs. There are also works which focus on style, namely generating gestures with the style of a specific person. Recently, there is also a work on learning the shape of iconic gestures based on the simplified shape of the object it represents.

Based on this state of the art, we identify three limitations and propose our research questions accordingly. The first research question is that we want to predict the timing of gestures, which means when to perform the ideational gestures and when to perform beat gestures. The second question is that we want to obtain the shape of metaphoric gestures. We will use the concept of image schema for this. The third question is that we want to investigate which word embedding technique is more suitable to represent the text for gesture generation problem.

Chapter 4

GestIS Corpus

We use the Gest-IS English corpus by Saint-Amand (2018). The Gest-IS corpus contains dialogues in English, Italian, and Bulgarian. However, we use only the English-language dialogues. All information we write here is only about the English-language part of the corpus. The speakers in the dialogues are university students of age between 22 and 26 years old. The dialogues themselves were recorded at the University of Edinburgh. The corpus consists of 9 dialogues of a dyad, a man and a woman, both of them are sitting, discussing various topics in a face-to-face setting. The total duration of the dialogues which we use is around 50 minutes. In those dialogues, the speakers are talking about the physical description of some places, the physical description of some people, scenes of two-person interactions, and instructions to assemble a wooden toy. There is one video (MOV format) per dialogue per speaker. In the video, both hands, the torso, and the face of the speaker are visible. On the other hand, there is only one audio recording per dialogue (WAV format), therefore the voices of the two speakers are mixed in the same audio file. However, the two speakers rarely interject each other. An example of the dialogue scene can be seen in Figure 4.1. The corpus has several layers of gesture annotations: communicative or non-communicative gestures, the gesture types, and the gesture phases. It should be noted that the annotations only take into account the hand movements, which means at least one hand must do the gesture. There are separate gesture annotations for the left hand and the right hand. The annotation was done by using ELAN [1]. The various annotations, as are displayed in ELAN can be seen Figure 4.2. The corpus also has transcription timestamp. The transcription is for each word and there are both starting and ending timestamps. On top of these readily available annotations, we also extract the eyebrow movements from the video files and the acoustic features from the audio files.

4.1 Communicative or Non-Communicative Gestures

The difference between communicative and non-communicative gestures is about whether the gesture has a communicative role in the dialog. Communicative gestures are related to the speech content or the rhythm of the speech. On the other hand, non-communicative gestures are movements which have no relationship with the speech. Examples of non-communicative gestures are scratching head or rubbing eyes. It should be noted that

[1] http://tla.mpi.nl/tools/tla-tools/elan/

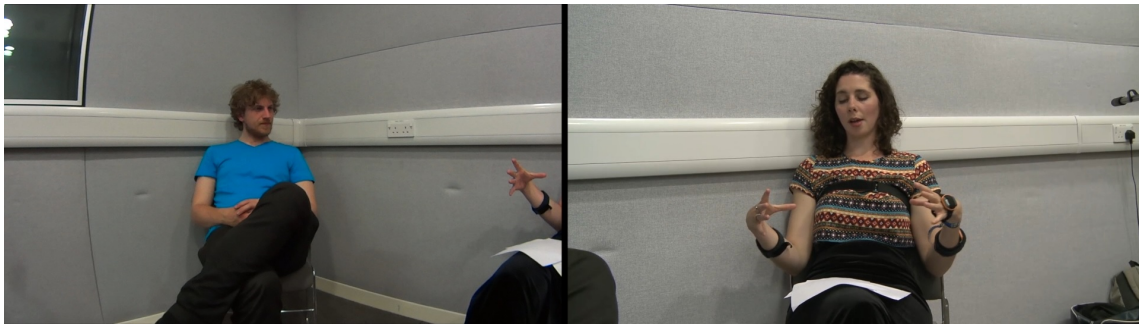

Figure 4.1 – The screenshot of one of the dialogues. We call the speaker on the left side as "Speaker A" and the speaker on the right side as "Speaker B"

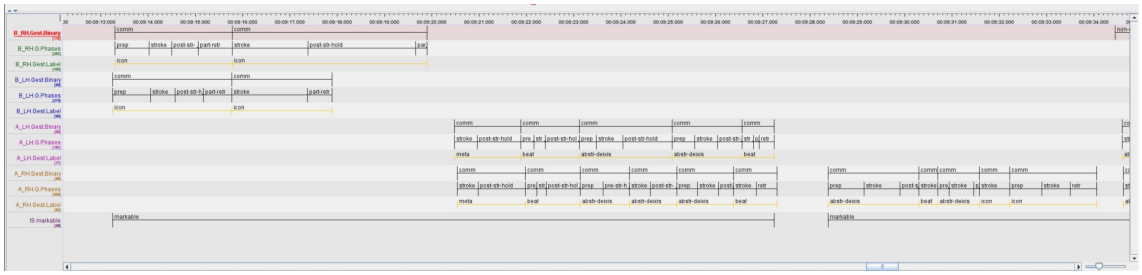

Figure 4.2 – The screen shot of the gesture annotation as is displayed in ELAN. A_ annotation refers to the speaker A while B_ refers to speaker B. _LH_ refers to the left hand while _RH_ refers to the right hand.

although the creator of this corpus calls these movements as non-communicative gestures, what we call as "gesture" in our work refers exclusively to what this corpus calls "communicative gesture".

4.2 Gesture Types

The gesture type annotation is based on the types as defined in McNeill (1992). This annotation only exists for "communicative gesture" as is defined in section 4.1. In this corpus, there are seven possible gesture types, namely iconic, metaphoric, concrete deixis, abstract deixis, nomination deixis, beat, and emblem. Unlike the standard McNeill's gesture classification, in this corpus the deictic is divided into three types, namely concrete deixis, abstract deixis, and nomination deixis.

Concrete deixis is a gesture which points to a concrete object or a location which exists in the vicinity of the speaker. For example, a gesture pointing to a nearby man accompanying an utterance "I saw this man" is a concrete deixis gesture because the gesture is pointing to an object which is in the vicinity of the speaker

Abstract deixis is a gesture which virtually points to an object or a location which does not exist in the vicinity of the speaker. For example, a pointing gesture which points toward nothing while accompanying an utterance "I saw this man" can be an abstract deixis gesture because the gesture only virtually points to the man: the speaker is imagining that the man he is talking about is in the direction he is pointing to. Similarly, a gesture of a hand moving to the right side while accompanying an utterance "there is a bedroom on the right" can also be an abstract deixis gesture because the bedroom he is talking about

is not physically on his right side. Instead, he is imagining that he is inside the apartment, and the bedroom he is talking about is on his right side.

Nomination deixis is a gesture where the index finger is extended to emphasize a word or a phrase. This gesture does not point to an object or location, neither physically nor virtually. An example of this is the gesture of an index finger pointing upward accompanying the utterance "Tom is the man who can help us" and the gesture coincides with the word "Tom". This gesture serves to put the emphasis at the mention of Tom.

4.3 Gesture Phases

The gesture phase annotation is based on the phases as defined by Kendon (1980). This annotation only exists for "communicative gesture" as is defined in section 4.1. In this corpus, there are seven possible gesture phases, namely preparation, pre-stroke hold, stroke, post-stroke hold, partial retraction, retraction, and recoil. It should be noted that in this corpus, beat gestures have phases like ideational gestures.

For the stroke phase, there is a problem on how to disambiguate a stroke which contains repetitive movement from multiple strokes. It should be remembered that multiple gestures can indeed be chained together. The disambiguation is done according to the movement uniformity. If the movements are symmetrical and uniform, then they belong to the same stroke. If the movements are not uniform, then they are different strokes. However, if the speed and movement direction change, then the slower one is a preparation of the next gesture while the rapid one is the stroke. For pointing gesture used as a concrete deixis gesture, the stroke is defined as the act of extending the finger instead of the act of holding the finger straight.

There is also a problem of disambiguating the hold phases. This is also because multiple gestures can be chained together. It is not always clear if a hold phase is a post-stroke-hold (i.e. a hold following the previous stroke) or a pre-stroke hold (i.e. a hold preceding the next stroke). In this case, the linguistic cues are used to disambiguate the two. Pre-stroke holds are more likely to accompany discourse connectives, pronouns, and temporal adverbial (e.g. "while" and "when"). Pre-stroke holds are also more likely to happen during hesitation pauses. On the other hand, post-stroke holds are more likely to happen during a fluent speech.

The phase annotation is done by a frame-by-frame analysis. This is done so that the speed difference and the movement can be observed more clearly. For example, when the hand is moving, it will appear blurred. On the other hand, when the hand is not moving, it will appear clear. This difference can be used to mark the hold phases.

It can be seen that the ambiguity problem in the phase annotation comes from the fact that consecutive gestures can be chained together. In the event of multiple movements in the stroke, it would have to be decided whether those multiple movements actually belong the one stroke. And in the event that there are two strokes with a hold in between, it would have to be decided whether that hold is a post-stroke hold of the first gesture or a pre-stroke hold of the second gesture.

4.4 Gesture Statistics

We count the number of each gesture type and we calculate their average duration (Tables 4.1, 4.2, 4.3, 4.4). We follow the segments given in the corpus (Figure 4.2). As in

Gesture	Segment Count	Average Duration (second)
Beat	46	0.849
Metaphoric	55	1.501
Iconic	56	2.179
Concrete Deixis	6	1.367
Abstract Deixis	22	2.177
Nomination Deixis	1	0.960
Emblem	4	1.770

Table 4.1 – The left hand gesture statistics of speaker A

Gesture	Segment Count	Average Duration (second)
Beat	38	0.807
Metaphoric	43	1.236
Iconic	91	2.026
Concrete Deixis	18	1.753
Abstract Deixis	33	1.863
Nomination Deixis	1	0.520
Emblem	3	2.120

Table 4.2 – The right hand gesture statistics of speaker A

the corpus, we differentiate the left hand gestures and the right hand gestures. We also differentiate the gestures according to the speaker.

We can observe in Tables 4.1, 4.2, 4.3, and 4.4 that beat gestures tend to be short. This is probably caused by the fact that beat gestures have simpler shapes. Unlike ideational gestures, beat gestures do not convey complex concept nor object. Surprisingly, the metaphoric gestures tend to be shorter than iconic gestures even though both of them can have complex shape. The metaphoric gestures are also more numerous. Probably this is caused by the topic of the conversations which is more about physical objects. We also observe that nomination deixis tend to be shorter than both the concrete deixis and abstract deixis. This is probably because nomination deixis is more like a beat gesture than a deixis gesture. Nomination deixis does not point to an object or location, but marks an emphasis. In this respect nomination deixis is more like beat gestures than deixis gestures. Indeed, we observe that the average duration of the nomination deixis are similar to the average duration of beat.

We also count the number of the strokes of the ideational gestures and we calculate their average duration (Tables 4.5, 4.6, 4.7, 4.8). It should be noted that the number of the strokes can be less than the number of the corresponding gesture segments (Tables 4.1, 4.2, 4.3, 4.4) because a gesture can be interrupted, and thus preventing the stroke from being executed. We also follow the stroke segments as given in the corpus (Figure 4.2). As in the corpus, we differentiate the left hand strokes and the right hand strokes. We also differentiate the strokes according to the speaker.

We observe in Tables 4.5, 4.6, 4.7, and 4.8 that the average stroke duration of nomination deixis is shorter than the others. Iconic gesture strokes tend to be slightly longer

Gesture	Segment Count	Average Duration (second)
Beat	93	1.008
Metaphoric	57	1.267
Iconic	69	2.005
Concrete Deixis	17	1.696
Abstract Deixis	82	1.608
Nomination Deixis	7	0.931
Emblem	2	2.220

Table 4.3 – The left hand gesture statistics of speaker B

Gesture	Segment Count	Average Duration (second)
Beat	42	1.027
Metaphoric	56	1.329
Iconic	69	2.009
Concrete Deixis	44	1.872
Abstract Deixis	37	1.666
Nomination Deixis	17	1.226
Emblem	2	2.180

Table 4.4 – The right hand gesture statistics of speaker B

Gesture	Stroke Count	Average Duration (second)
Metaphoric	47	0.467
Iconic	50	0.486
Concrete Deixis	6	0.420
Abstract Deixis	22	0.489
Nomination Deixis	1	0.280

Table 4.5 – The left hand ideational gesture stroke statistics of speaker A

Gesture	Stroke Count	Average Duration (second)
Metaphoric	39	0.414
Iconic	86	0.765
Concrete Deixis	16	0.380
Abstract Deixis	31	0.512
Nomination Deixis	1	0.240

Table 4.6 – The right hand ideational gesture stroke statistics of speaker A

Gesture	Stroke Count	Average Duration (second)
Metaphoric	55	0.498
Iconic	66	0.695
Concrete Deixis	15	0.443
Abstract Deixis	74	0.555
Nomination Deixis	6	0.333

Table 4.7 – The left hand ideational gesture stroke statistics of speaker B

Gesture	Stroke Count	Average Duration (second)
Metaphoric	56	0.444
Iconic	67	0.596
Concrete Deixis	42	0.435
Abstract Deixis	33	0.559
Nomination Deixis	17	0.409

Table 4.8 – The right hand ideational gesture stroke statistics of speaker B

than the others, but the difference is not much. The ideational gesture strokes other than the nomination deixis seem to have similar average durations.

4.5 Eyebrow Movement Extraction

We use OpenFace (Baltrusaitis et al. (2018)) to extract the eyebrow movements. OpenFace extracts facial movements, encoded by using Facial Action Coding System (FACS) (Friesen and Ekman (1978)). FACS divides a facial movement into the constituent movements. Each constituent movement is called Action Unit (AU).

OpenFace works in two steps. Firstly, it detects the facial landmark points from the video and track them. Secondly, OpenFace detects the Action Units's intensity and presence. The step of detecting and tracking the facial landmark points is performed by a neural network model called Convolutional Experts Constrained Local Model (CE-CLM) (Zadeh et al. (2017)). This first step is important because before detecting the Action Units, it is necessary to locate the face itself and the facial landmarks. Those facial landmarks can be seen as the dotted areas in Figure 4.3. The second step is then performed by a Support Vector Machine based technique from Baltrušaitis et al. (2015) to detect the presence and intensity of the Action Units.

OpenFace yields the presence and intensity of all the Action Units. Sometimes, some Action Units are not present. This is expected. For example, when the eyebrow is not visible, the eyebrow-related Action Units should be marked to be not present. There is also a confidence rate which signifies how confident OpenFace is on its reading.

We apply a two-step filter to clean the data. The first step is eliminating those whose confidence rate is below 0.85. Basically, we remove the readings which we are not sure about. In the second step, we take the consecutive values where the Action Unit is always present as one segment, and we eliminate the entire segment if their average intensity is below 1. Basically, we remove the readings whose intensity is too low.

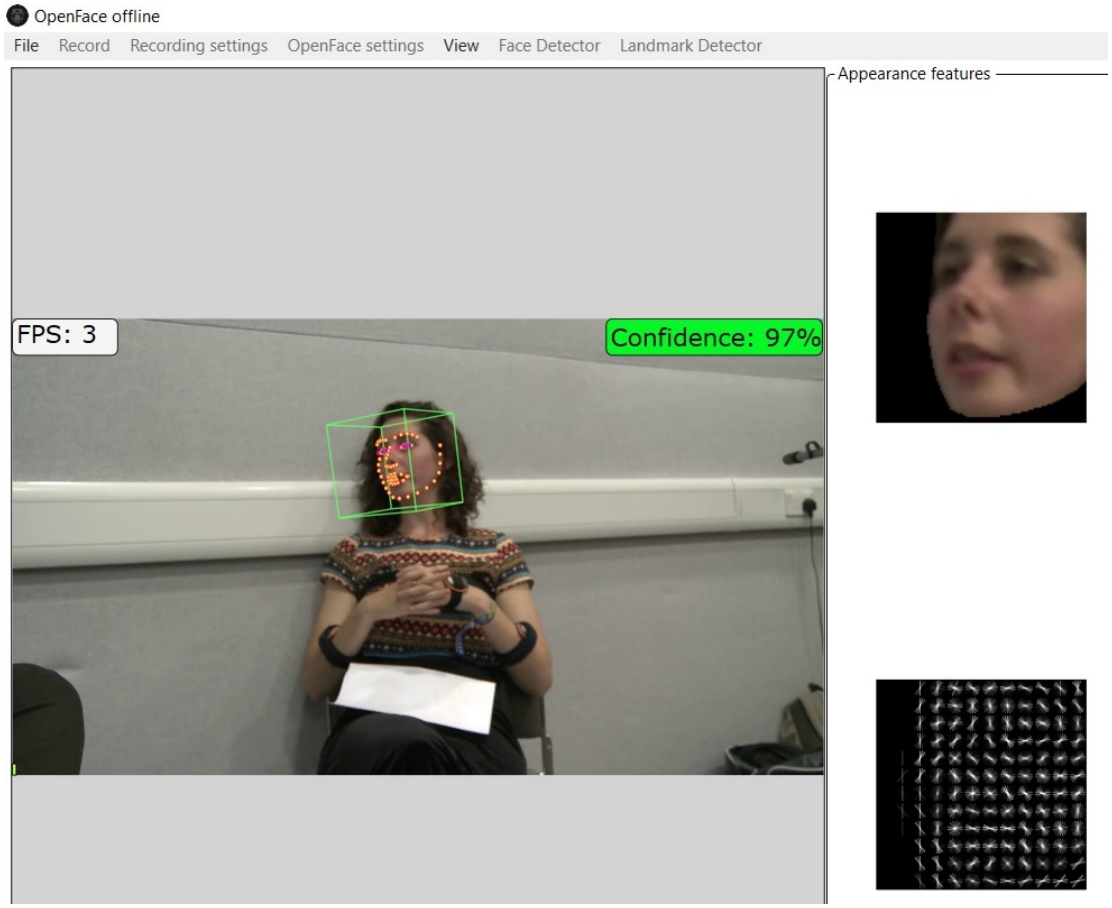

Figure 4.3 – OpenFace's facial landmark point tracking

AU01	AU02	AU04	Segment Count	Average Duration (second)
✓	✗	✗	8	1.835
✗	✓	✗	8	0.355
✗	✗	✓	7	1.606
✓	✓	✗	13	1.302
✗	✓	✓	14	0.983
✓	✗	✓	15	1.728
✓	✓	✓	19	1.465

Table 4.9 – The speaker A's eyebrow movement statistics. The combination of different AUs indicate a union

AU01	AU02	AU04	Segment Count	Average Duration (second)
✓	✗	✗	8	2.140
✗	✓	✗	8	2.275
✗	✗	✓	8	15.580
✓	✓	✗	13	2.071
✗	✓	✓	15	9.427
✓	✗	✓	13	10.262
✓	✓	✓	18	7.951

Table 4.10 – The speaker B's eyebrow movement statistics. The combination of different AUs indicate a union

There are three action units (AUs) which represent eyebrow movements, namely AU1 (inner brow raiser), AU2 (outer brow raiser), and AU4 (brow lowerer). The presence of either AU1 or AU2 represents rising eyebrow while the presence of AU4 represents lowering eyebrow.

4.6 Eyebrow Movement Statistics

The corpus does not have the eyebrow movement annotation. However, the corpus has the conversation videos where the speaker's face is visible. The eyebrow movement information can therefore be extracted from those videos. These eyebrow movements are represented by Action Units (AUs) 1, 2, and 4. We calculate the statistics of those Action Units (Tables 4.9, 4.10). For this data, the notion of segment we use simply means a continuous activation of the Action Unit in question. Several action units can be active concurrently, therefore we present the statistics of the union of the activation of different action units. Following the gesture statistics, we split the statistics according to the speaker.

We observe in Tables4.9 and 4.10 that between speakers A and B, the number of segments are similar, but the average durations is much higher on speaker B. For speaker A, we observe that the average duration of AU2 (outer eyebrow raiser) is much shorter than AU1 or AU4. For speaker B, we observe that the average duration of AU4 (i.e. eyebrow lowerer) is much longer than the average duration of AU1 or AU2. Interestingly, despite

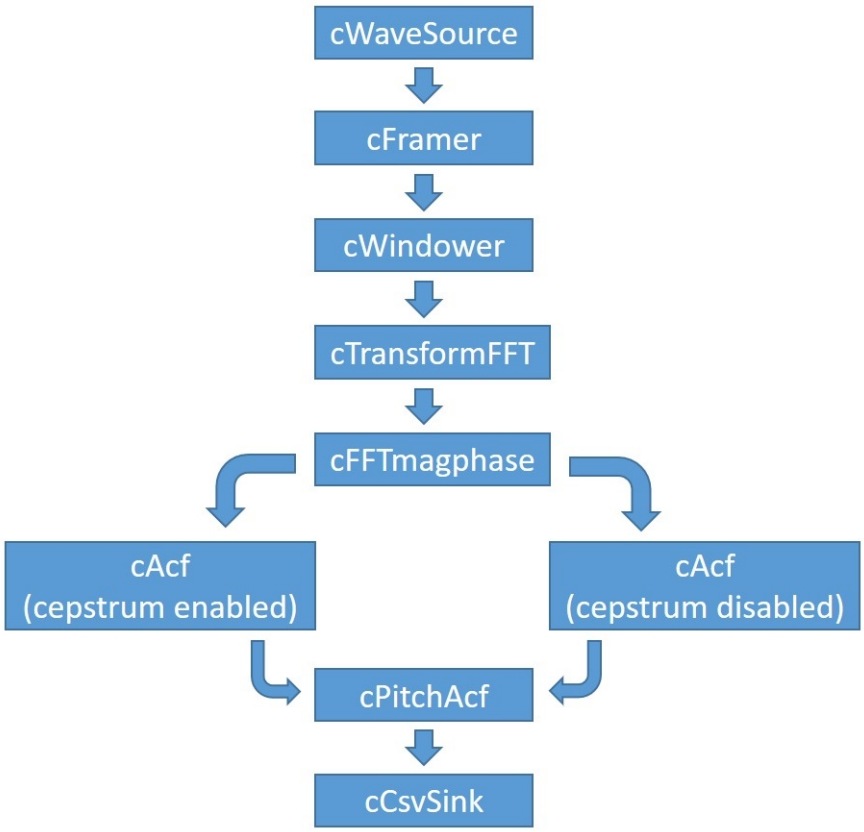

Figure 4.4 – OpenSmile's pipeline to extract F_0. cWaveSource is the component to read the WAV audio file and cCsvSink is the component to write the output into a CSV file. cPitchAcf is the component which extracts the F_0

of the relationship between beat gestures and eyebrow movements (see Section 2.4), we do not find any concurrence of the eyebrow movements and beat hand gestures.

4.7 Acoustic Features Extraction

Although audio processing has a long history in computer science, audio data has to be processed first to extract the useful features. We use OpenSmile (Eyben et al. (2010)) to extract those audio features. OpenSmile is used by creating a pipeline of component. Each component takes certain features as the input and yields certain other features as the output. There are also special components which read the audio file from the persistent storage or write the extracted features as a file to the persistent storage. These components can be chained together so that we can extract the features we want from the audio file. For example, if we want to extract the pitch features, the component which reads the pitch features need the output of the autocorrelation component. An example of such pipeline is in Figure 4.4.

we use OpenSmile to extract the audio features with 100 milliseconds time-step. We choose three prosody features, fundamental frequency (F_0), F_0 direction score, and intensity, for their temporal relation with gestures (Loehr (2012); Cravotta et al. (2019)). We also extract the Mel-frequency cepstral coefficients (MFCC), which for each time step

is represented as a 13-dimensional vector. The Mel-frequency cepstral coefficients themselves have been successfully used to generate body movements from the speech accoustic (Hasegawa et al. (2018); Kucherenko et al. (2019)).

4.8 Conclusion

Gest-IS corpus Saint-Amand (2018) contains face-to-face dialogues of a dyad. There are separate videos of each speaker, but the audio recording is mixed. The corpus has several gesture annotations. The first one is communicative or non-communicative gestures, which tells whether the movement is for a communication purpose. The second one is the gesture type, which is based on McNeill (1992)'s classification. The third one is the gesture phase, which is based on the phase classification by Kendon (1980). The annotation for the left hand and the right hand are separate. The corpus also includes the transcript of the conversation.

We also calculate some statistical measures of the data. We calculate the number of segments and the average duration of the gestures and the strokes. We separate the data according to the hand and the speaker. We also calculate the eyebrow movement statistics. The corpus does not include eyebrow movement in its annotation, but we extract it ourselves. We calculate the number of segments and the average duration of the eyebrow movements with different combinations of the Action Units.

Chapter 5

Prediction of Gesture Timing

In this work, we attempt to tell when a virtual agent should perform different types of gestures. In another word, we learn the gesture timing. This includes the timing of when no gesture is performed. We distinguish beat gestures from ideational gestures. Additionally, for the ideational gestures, we also distinguish the stroke phase from the other phases.

This work can be seen as the bridge between the acoustics-based generators (see Section 3.5) and the semantics-based generators (see Section 3.4) because beat gestures are related to the speech's rhythm while ideational gestures are related to the speech's semantics. Although it can be argued that a technique which learns the body movements from both the text and the acoustics (see Section 3.6) also implicitly learns the timing, there is also a benefit from separating the learning of the timing from the learning of the shape. First, when the entire movements are generated together, it might not be clear which segments are supposed to represent the semantics and which segments are supposed to mark the rhythm or discourse context. Secondly, it also enables a different model which computes the gesture shape, which can be developed separately, to be plugged in.

We also propose an objective evaluations metric based on a sequence comparison technique which tolerates shift and dilation to a certain extent. The underlying spirit is similar to the other works which allow many-to-many relationships (e.g. Ginosar et al. (2019); Ferstl et al. (2019); Wu et al. (2021)): for each input, there can be multiple correct outputs. However, our approach is more narrow: we only tolerate differences in the form of shifts and dilations.

5.1 Feature Extraction

We use the Gest-IS English corpus (see Chapter 4). We divide the communicative gestures into beat and ideational (i.e. everything other than beat). We distinguish beat gestures from ideational gestures because ideational gestures convey a specific meaning while beat gestures mark the speech rhythm and discourse context. Moreover, beat gestures tend to appear at the start of the topic while ideational gestures (Cassell et al. (2001)) tend to appear in the content of the topic where the new information is carried (Halliday (1973)). We also divide the ideational gestures according to their phases into strokes and the other phases because the stroke phase is known to be where the meaning is conveyed and is

usually near the pitch accent (Kendon (1980)). Our usage of gesture phases as the classes bears some similarity with Ferstl et al. (2019) (see Section 3.5).

As such, we classify the gestures into four classes:

- "NoGesture" refers to the instance when the person does not perform a gesture.
- "Beat" refers to the instance when the person does beat gesture.
- "IdeationalOther" refers to the instance when the person does a non-stroke phase (e.g. preparation, retraction) of an ideational gesture.
- "IdeationalStroke" refers to the time when the person does the stroke phase of an ideational gesture. Note that beat gestures have neither stroke nor non-stroke phase (McNeill (1992)).

The model uses only the acoustic as the model input. We extract the fundamental frequency (F_0), the F_0 direction score, intensity, and the Mel-frequency cepstral coefficients (MFCC) where each timestep is 100 ms. The extraction process is explained in Section 4.7. We decompose the speech into utterances where an utterance is defined as a sequence of words surrounded by pauses. One utterance is one sample. To define the utterance boundaries, we use the concept of Inter-Pausal Unit (IPU) (Levitan and Hirschberg (2011)): two consecutive utterances are separated by a silence of at least 200 milliseconds long (Peshkov et al. (2013)).

We also extract eyebrow movements because we will also investigate if inclusion of eyebrow movements helps to make the prediction better. We extract AU1 (inner brow raiser), AU2 (outer brow raiser), and AU4 (brow lowerer). The extraction process is explained in Section 4.6.

In our full dataset, we have 4161 time-steps of "NoGesture"s (34.07%), 1106 time-steps of "Beat"s (9.06%), 4208 time-steps of "IdeationalOther"s (34.45%), 2739 time-steps of "IdeationalStroke"s (22.42%). The duration of each time-step is 100 milliseconds. In total, we have 798 samples.

5.2 Model

5.2.1 Problem Statement

Let X be the input and Y be the output. Both X and Y are sequences with the same length. Onward, we will refer to their length as l. X is a sequence of vector. Let X_i be the vector at timestep i, X_i is a 3-dimension vector of real numbers containing the fundamental frequency (F_0), the F_0 direction score, and the intensity. Y is a sequence of gesture class (Formulae 5.3 and 5.2).

$$X_i = (F_0, F_0\ direction, intensity) \in \mathbb{R}^3, \tag{5.1}$$

$$CLASSES = \langle \text{“}NoGesture\text{”}, \text{“}Beat\text{”}, \text{“}IdeationalOther\text{”}, \text{“}IdeationalStroke\text{”} \rangle \tag{5.2}$$

$$Y_i \in CLASSES \tag{5.3}$$

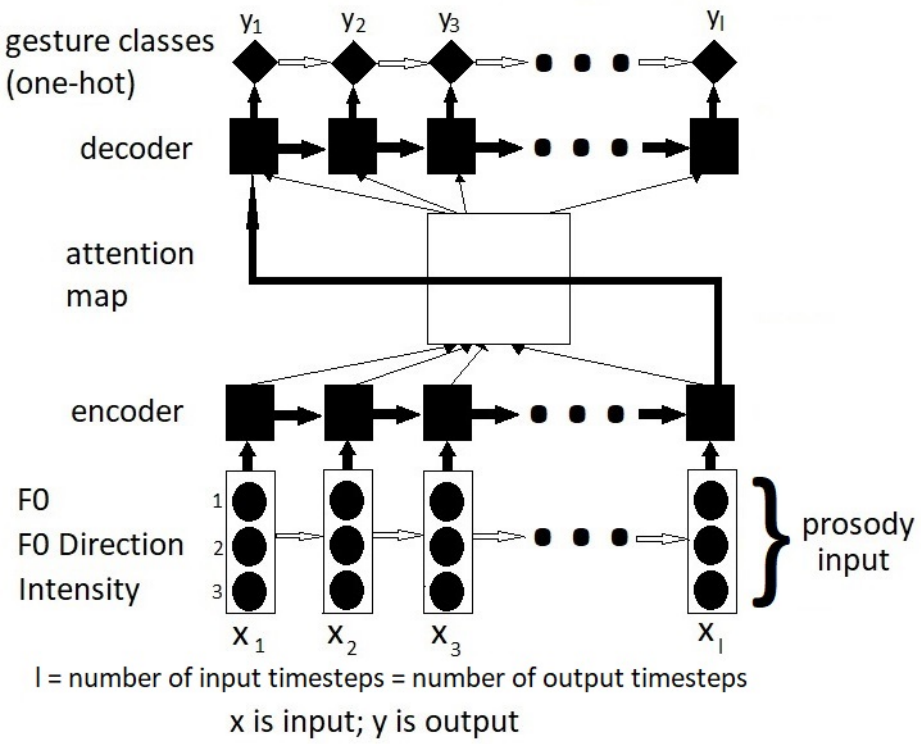

Figure 5.1 – The Neural Network Model

5.2.2 Model Overview

We use recurrent neural network with attention mechanism (Bahdanau et al. (2014)) to perform the prediction. The recurrent neural network with attention mechanism is an extension of the encoder-decoder model. The standard encoder-decoder model compresses the entire information from the input sequence into a fixed-length vector at the last encoder node. The attention mechanism adds an attention map between the encoder and the decoder. The map itself is a neuron matrix of the size l^2. If w_{ij} is the weight in the attention map at position $\langle i, j \rangle$, then w_{ij} represents the weight of the input at timestep i on the output at timestep j. This neuron matrix enables focusing the "attention" toward some specific input timesteps. If the input at timestep i is pertinent on the output of timestep j, then the w_{ij} would be high. Those weights are learned during the training, similar to all other weights in the network. Because this is a multi-class classification problem where the output of each timestep belongs to one of the gesture classes (Formula 5.3 and 5.2), we use a one-hot encoding to encode Y_i. We present the schema of the model in Figure 5.1.

The model needs all the inputs to be of the same length. The model also needs all the outputs to have the same length. However, our samples are natural utterances which have different lengths. Thus, we pad those sequences to make them have the same length. We pad the inputs with 0-vectors and we pad the outputs with the "suffix" auxiliary class. Effectively, we modify Formula 5.2 by adding the "suffix" class. In our full dataset, after we add the "suffix" class, the distribution of the data is as the following:

- NoGesture: 4161 time-steps (6.14%)
- Beat: 1106 time-steps (1.63%)

- IdeationalOther: 4208 time-steps (6.20%)
- IdeationalStroke: 2739 time-steps (4.04%)
- Suffix (auxiliary class): 55616 time-steps (81.99%)

5.2.3 Implementation details

We implement the model by using the Zafarali [1]'s code as the template. The code itself is written in Keras [2]. We replace the input of the original code [3] with the input we describe in Sub-Section 5.2.1 and the modification to make the samples to have the same length as explained in Section 5.2.2. We use categorical cross-entropy as the loss function. Categorical cross-entropy is a loss function which can be used for a classification problem where there are multiple possible classes but every sample belongs to exactly one class. It works by calculating the entropy of the probabilities of the different classes, and therefore the loss is reduced when there is less uncertainty of the class which the sample belongs to. We use Adam (Kingma and Ba (2014)) as the optimization method. Adam is an optimization method where the learning rate changes between each parameter and also changes during the training process. To deal with the class imbalance in the dataset, we assign low weights to frequently-occurring classes and high weights to rarely-occurring classes.

5.3 Evaluation Measure

Our work uses encoder-decoder model. The prior works which also use encoder-decoder model use domain specific measurements to evaluate the performance of their model. Sutskever et al. (2014), the pioneer of the seq2seq formulation, use BiLingual Evaluation Understudy (BLEU) to evaluate their language translator. Chorowski et al. (2015) use phoneme error rate (PER) to evaluate their speech recognition model. Bahdanau et al. (2016) use Character Error Rate (CER) and Word Error Rate (WER) to evaluate their speech recognition model.

There is not always a gesture on every pitch accent. Moreover, gesture stroke may precede the prominent speech part. To measure the error rate of our model, we need a sequence comparison technique to quantify the similarity between the ground truth and the prediction: this technique should tolerate shifts and dilations to some extent. Tolerating the shift means that the technique must tolerate to a certain extent that the matching blocks can start or end at different times. Tolerating the dilation means that the technique must tolerate to a certain extent that the matching blocks can have different lengths up to a certain extent. Practically, it means that we tolerate if the predicted segment starts or ends slightly earlier or slightly later than the ground truth. We also tolerate if the predicted segment is slightly shorter or longer than the ground truth. For example, in Figure 5.2, we see that the predicted "IdeationalStroke" starts 100 ms earlier and ends 100 ms later than in the ground truth. The predicted "IdeationalStroke" is also 200 ms longer.

Dynamic Time Warping (Bellman and Kalaba (1959)) is a sequence comparison technique which tolerate shifts and dilations. However, this technique does not have a continuity constraint. That is, two consecutive elements which belong to the same class in a

[1] https://github.com/datalogue/keras-attention

[2] https://keras.io/

[3] Originally for date format translation (e.g. the input is "Saturday 9 May 2018" string and the output is "2018-05-09" string)

sequence might be matched against two non-consecutive elements in the other sequence. Without the continuity constraint, we might end up with a match like in Figure 5.3. In that figure, we can see that the "NoGesture" elements in the middle of the ground truth are matched with the "NoGesture" elements in the prediction before and after the "IdeationalStroke". However, a continuous "NoGesture" is different from a "IdeationalStroke" preceded and followed by "NoGesture".

Thus, we propose a sequence comparison technique to quantify the similarity between the ground truth and the prediction where a block of consecutive elements with the same class is matched against a block of consecutive elements of that class. We use this technique to evaluate our result.

Our measurement technique uses the sequence comparison algorithm proposed by Dermouche and Pelachaud (2016). It measures the city-block distance between a block in the ground truth and a block in the prediction. This distance metric tolerates shift and dilation up to a certain threshold. If the distance between the two blocks is below the threshold, then they are considered to be aligned. The alignment formula is shown at Formula 5.4. We define b_{ps} and b_{pe} respectively as the start and the end of the prediction block. Correspondingly, we define b_{ts} and b_{te} respectively as the start and the end of the ground truth block. We also define T as the distance threshold. We define the alignment condition between the prediction block and ground truth block in Formula 5.4.

$$ALIGNED \Longleftrightarrow |b_{ps} - b_{ts}| + |b_{pe} - b_{te}| \leq T \tag{5.4}$$

We measure the alignment based on how many blocks are aligned and we normalize it against the lengths of those blocks and the frequency of that particular class. At its essence, we try to find out for how many time-steps the prediction is aligned to the ground truth, subject to the condition that consecutive time-steps in the ground truth which share the same class must be matched to consecutive or the same time-steps in the prediction which belong to that class as well. This is then normalized against the frequency of that class.

We also introduce the concept of "insertion" and "deletion". A block which exists in the prediction but has no match in the ground truth is considered to be "inserted". This is conceptually similar to *false positive*: we predict what actually does not happen. The block exists in the prediction but it does not exist in the ground truth. Similarly, a block which exists in the ground truth but has no match in the prediction is considered to be "deleted". This is similar to *false negative*: we fail to predict something which actually happens. For example, in Figure 5.4, we observe an "inserted" "NoGesture" block and a "deleted" "IdeationalOther" block. The precise definition of alignment, insertion, and deletion score are at Formulae 5.5. The meaning of the symbols are furnished in Table 5.1. Basically, a_c is the proportion of class c which is aligned, d_c is the proportion of class c which is deleted, and i_c is the proportion of class c which is inserted.

The ideal alignment score is 1 while the ideal deletion and insertion score are 0. It means everything is aligned and there is neither deleted nor inserted block. The insertion score of class *c* can exceed 1 if we predict class *c* more frequently than it actually occurs. On the other hand, the deletion score is always between 0 and 1. The deletion score of class *c* is 1 when we fail to predict any of the block of that class. For the alignment score, if the predictor is accurate but slightly overestimates the length of the block, then the alignment score will be slightly higher than 1. On the other hand, if the predictor is accurate but often slightly underestimates the length of the block, then the alignment score will be slightly lower than 1.

Symbol	Meaning
n	number of samples in the dataset
t_c	number of timesteps of class *c* in the dataset
p_c	proportion of class *c* in the dataset
l	sample length (the same for all samples)
$b.d$	deleted block
d_c	deletion score of class *c*
$b.i$	inserted block
$b.p$	predicted block
$b.t$	ground truth block
a_c	alignment score of class *c*

Table 5.1 – Symbols at Formulae 5.5

$$
\begin{aligned}
p_c &= \frac{t_c}{n \times l} \\
d_c &= \frac{\Sigma_{b.d} length(b.d)}{n \times l \times p} \\
i_c &= \frac{\Sigma_{b.i} length(b.i)}{n \times l \times p} \\
a_c &= \frac{\Sigma_{(b.p,b.t).aligned}(length(b.p) + length(b.t))}{2 \times n \times l \times p}
\end{aligned}
\tag{5.5}
$$

5.4 Objective Experiment

To evaluate our model, including the pertinence of the input variables, we perform 7 objective experiments. Experiment 1 is for obtaining the baseline performance by generating random outputs according to the data distribution. Experiment 2 is for obtaining the performance of the neural network. Experiment 3 is an ablation study to find out which features are more pertinent. Experiment 4 is for finding out the effect of including eyebrow movements on the "Beat" class. Experiment 5 is for finding out the whether using Mel-frequency cepstral coefficients as the input features to infer the gesture class works better. Experiment 6 is for finding out the effect of using both Mel-frequency cepstral co-

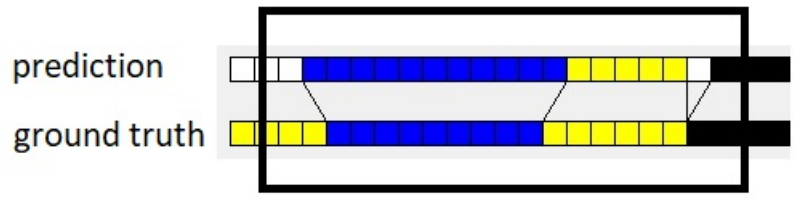

Figure 5.2 – Alignment Example. Each cell is 100 ms long.
Blue: "IdeationalStroke"
Yellow: "'IdeationalOther"

efficients and prosody features as the inputs. Experiment 7 is for finding out whether a model trained with one speaker only is generalizable to the dialogue counterpart.

In Experiments 1, 2, 3, 4, 5, and 6, we break the full data set into training, validation, and testing sets in exactly the same way. Thus, if a particular utterance goes to the testing data set in one experiment, it also goes to the testing data set in the other experiments. We mix all samples from all videos from both speakers and then we randomly split our data with the proportion of 64% training data, 16% validation data, and 20% testing data. This is chosen according to the common 80/20 rule. 80% of the data is for both training and validation and 20% of the data is for testing. The 80% is then split again $80\% \times 80\% = 64\%$ for training and $80\% \times 20\% = 16\%$ for validation. Therefore, each of the training, validation, and testing dataset contains a mix of samples from both speakers and different dialogues. Experiment 7, by its nature, requires us to separate the dataset into two halves, namely when the first person is speaking and when the second person is speaking. We use 80% of the data from one speaker as the training data set, 20% of the data from the same speaker as the validation data set, and 100% of the data from the other speaker as the testing data set.

To make the results on the neural network models comparable, we expend equivalent "effort" on the experiments where we train neural network models. We randomly vary the encoder and decoder dimensions from 1 up to the number of features: 3 with prosody features, 13 with Mel-frequency cepstrum features, 16 with both prosody and Mel-frequency cepstral coefficient features. We run 25 trainings with 500 epochs, 25 trainings with 1000 epochs, and 5 training with 2000 epochs. Therefore, in total we have 55 models for each problem. We choose the model with the best performance on the validation data set. Specifically, we use the weighted average of "Beat Alignment", "IdeationalStroke Alignment", "IdeationalOther Alignment", and "IdeationalOther Alignment" scores, but we set a constraint that for each of them the score must be at least 0.05. The weights are based on the frequency of those classes in the data set. A challenge we face is that the loss function used in the training concerns only the matches at the same timestep, therefore ignoring the possibilities of shifts or dilations, which means that the network is not completely optimized for our objective. Therefore, we have to rely on the stochasticity of the neural network. This situation triggers a question on whether the performance we see with the validation data set is a reliable proxy of what we will see when we use the testing data set.

On top of doing the training-validation-testing protocol, we also investigate the reliability of the training and the validation for each class. In the standard machine learning protocol, we train the model several times with different parameters or configurations, run each of the trained models on the validation data set, and choose the trained model with a good performance on the validation data set. In the training phase, the weights in

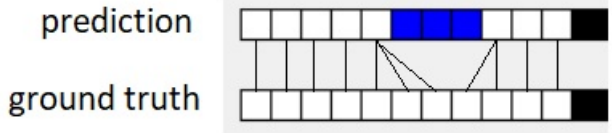

Figure 5.3 – Discontinuity Example. Each cell is 100 ms long.
White: "NoGesture"
Blue: "IdeationalStroke"

the network are automatically tuned so that the network learns the pattern of the training data. This training phase is stochastic, therefore every training session yields a different model. Because of the stochasticity, we do the training several times, and thus we get several potential models. In the next phase, which is the validation phase, we validate the models we get from the training against the validation dataset. This validation data set is not used in the training. We choose the model which performs the best against the validation dataset as the final choice. This protocol is based on two assumptions. The first assumption is that in the training phase, the resulting models will learn the pattern observed in the training data. Therefore, the learned models should perform better than chance against similar datasets. This is especially relevant for our case because during the training we optimize the networks against a loss function which compares the prediction against the ground truth at the same time step while we evaluate our models on a different metric which tolerates shift and dilation. The second assumption is that, when we choose among the trained models in the validation phase, a better performance against the validation dataset is a reliable proxy on the performance on the testing data set. Therefore, the performance of the models against the validation dataset should correlate with the performance of the models on the testing dataset.

In **Experiment 1 (random output)**, we generate random outputs according to the probability distribution of the gesture classes, while completely ignoring the prosody input. Specifically, we measure two sets of probabilities, namely the probabilities that a sample is started by a particular class and the probabilities that a class follows another (or the same) class. This is done because our data consists of sequences, where each element affects the next element. We match this result against the output from our ground truth. We do this 55 times and we measure the average of their performances. This can be seen as an extremely simple predictor and thus can be seen as the baseline result.

In **Experiment 2 (using neural network with the entire dataset)**, we build a neural network model, and then we do the training, validation, and testing. We also measure the training and validation reliability of the trained models.

In **Experiment 3 (ablation study)**, we want to observe how the three different prosody features (i.e. F_0, F_0 direction score, and intensity) affect the performance of the model. In order to do that, we use the model used in Experiment 2, but we replace some or all of the input features with random values. This way, we eliminate all information from those features and we force the model to rely only on the remaining features.

In **Experiment 4 (inclusion of eyebrow movements)**, we investigate whether inclusion of eyebrow movements helps on predicting beat class. Eyebrow movements often mark speech prosody and are aligned with pitch accent (Bolinger (1989); Ekman (1979)). Therefore, we include the eyebrow movements in the "Beat" class and we compare the per-

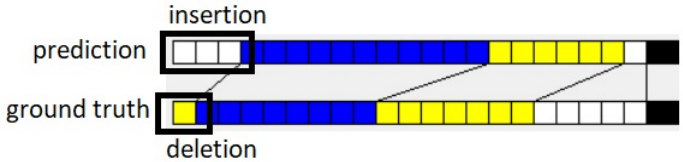

Figure 5.4 – Insertion and Deletion Example. Each cell is 100 ms long.
White: "NoGesture"
Yellow: "'IdeationalOther"
Blue: "IdeationalStroke"

Class	Alignment	Insertion	Deletion
Beat	**0.009**	0.936	0.990
IdeationalStroke	0.084	0.485	0.904
IdeationalOther	0.109	0.563	0.882
NoGesture	**0.533**	0.940	0.453

Table 5.2 – Exp 1: Random output result

Class	Alignment	Insertion	Deletion
Beat	**0.194**	3.127	0.802
IdeationalStroke	**0.507**	0.485	0.582
IdeationalOther	**0.304**	0.226	0.671
NoGesture	0.567	0.554	0.398

Table 5.3 – Exp 2: Using neural network with the entire dataset

formance of the network when the data includes only hand movements (i.e. Experiment 2), when the data considers hand movements and upward eyebrow movements (AU1 or AU2), and when the data considers hand movements and both upward and downward eyebrow movements (AU1 or AU2 or AU4). We measure the alignment, insertion, and deletion scores of the "Beat" class. We also measure the training and validation reliability.

In **Experiment 5 (Mel-frequency cepstral coefficients as input)**, we use the Mel-frequency cepstral coefficients instead of the three prosody features we use in Experiment 2 (i.e. F_0, F_0 direction score, and intensity) as the input features for our neural network. We measure the performance and the training and validation reliability.

In **Experiment 6 (both Mel-frequency cepstral coefficients and prosody as input)**, we use both the Mel-frequency cepstral coefficients and the three prosody features we use in Experiment 2 (i.e. F_0, F_0 direction score, and intensity) as the input features for our neural network. We measure the performance and the training and validation reliability.

In **Experiment 7 (trained with one speaker, tested on the other speaker)**, we train the model with one speaker of the dyad in our corpus and test it on the other speaker, and then we do the reverse. It should be noted that one speaker is a man and the other one is a woman.

Alignment Score of ...	Mean at Validation Data	Mean at Testing Data	Correlation
Beat	**0.202**	**0.244**	**-0.037**
IdeationalStroke	0.317	0.361	0.875
IdeationalOther	0.202	0.274	0.809
NoGesture	0.537	0.546	0.679

Table 5.4 – Exp 2: Using neural network with the entire dataset
Training and validation reliability

All features are randomized			
Class	Alignment	Insertion	Deletion
Beat	0.040	0.643	0.929
IdeationalStroke	0.038	0.072	0.952
IdeationalOther	0.025	0.027	0.960
NoGesture	0.347	0.275	0.641
Using intensity only			
	Alignment	Insertion	Deletion
Beat	0.0	0.786	1.000
IdeationalStroke	0.077	0.063	0.922
IdeationalOther	0.039	0.040	0.936
NoGesture	0.376	0.298	0.589
Using F_0 and the F_0 direction score only			
Class	Alignment	Insertion	Deletion
Beat	**0.175**	2.444	0.802
IdeationalStroke	**0.481**	0.503	0.563
IdeationalOther	**0.313**	0.179	0.637
NoGesture	**0.596**	0.555	0.379
Using F_0 only			
Class	Alignment	Insertion	Deletion
Beat	**0.179**	2.540	0.802
IdeationalStroke	**0.521**	0.515	0.553
IdeationalOther	**0.273**	0.155	0.664
NoGesture	**0.577**	0.570	0.393
Using F_0 direction score only			
Class	Alignment	Insertion	Deletion
Beat	0.044	0.548	0.929
IdeationalStroke	0.024	0.083	0.965
IdeationalOther	0.019	0.013	0.969
NoGesture	0.379	0.311	0.630

Table 5.5 – Exp 3: Ablation study

Condition	Alignment	Insertion	Deletion
Hand Only	**0.194**	3.127	0.802
With Upward Eyebrow Movement	0.136076	1.037975	0.829114
With Upward/Downward Eyebrow Movement	**0.222**	0.280	0.774

Table 5.6 – Exp 4: Inclusion of eyebrow movements

Training and validation reliability of the "Beat" alignment score			
Alignment Score of ...	Mean at Validation Data	Mean at Testing Data	Correlation
Hand Only	0.202	0.2444	-0.037
With Upward Eyebrow Movement	0.078	0.102	**0.414**
With Upward/Downward Eyebrow Movement	0.226	0.219	**0.925**

Table 5.7 – Exp 4: Inclusion of eyebrow movements

Class	Alignment	Insertion	Deletion
Beat	0.171	2.619	0.849
IdeationalStroke	**0.166**	0.977	0.855
IdeationalOther	0.362	0.538	0.652
NoGesture	0.440	0.789	0.551

Table 5.8 – Exp 5: Mel-frequency cepstral coefficients as input

Alignment Score of ...	Mean at Validation Data	Mean at Testing Data	Correlation
Beat	0.060	0.084	**-0.056**
IdeationalStroke	0.248	0.256	0.405
IdeationalOther	0.283	0.340	0.502
NoGesture	0.452	0.467	0.204

Table 5.9 – Exp 5: Mel-frequency cepstral coefficients as input
Training and validation reliability

Class	Alignment	Insertion	Deletion
Beat	0.000	2.429	1.000
IdeationalStroke	0.388	0.790	0.640
IdeationalOther	0.362	0.584	0.613
NoGesture	0.441	0.891	0.563

Table 5.10 – Exp 6: Both Mel-frequency cepstral coefficients and prosody as input

Alignment Score of ...	Mean at Validation Data	Mean at Testing Data	Correlation
Beat	0.080	0.0745	**0.025**
IdeationalStroke	0.272	0.265	0.472
IdeationalOther	0.302	0.351	0.622
NoGesture	0.425	0.465	0.386

Table 5.11 – Exp 6: Both Mel-frequency cepstral coefficients and prosody as input
Traiing and validation reliability

Trained on the 1st speaker, tested on the 2nd speaker			
Class	Alignment	Insertion	Deletion
Beat	0.015	1.049	0.982
IdeationalStroke	**0.506**	1.142	0.559
IdeationalOther	0.367	0.359	0.575
NoGesture	0.517	0.441	0.459
Trained on the 2nd speaker, tested on the 1st speaker			
Class	Alignment	Insertion	Deletion
Beat	0.132	3.679	0.856
IdeationalStroke	**0.396**	0.846	0.650
IdeationalOther	0.217	0.221	0.746
NoGesture	0.538	0.589	0.424

Table 5.12 – Exp 7: Trained with one speaker, tested on the other

5.5 Objective Experiment Results

In the performance of the model which is trained and tested with the entire data (Experiment 2, Table 5.3), we observe that the alignment scores outperform the random output (Table 5.2) on all classes. However, the alignment score of the "NoGesture" class is only slightly higher than the corresponding score of the random output. In the Table 5.4, where we run all the trained models against the validation dataset and against the testing dataset, we also observe that the mean alignment scores are indeed higher than the alignment scores of the random outputs. This suggests that the training process works because the models yielded by the training process indeed learn the pattern of the data. However, the correlation of the alignment scores on the validation dataset and testing dataset is only -0.037 on the "Beat" class, which is close to zero. That means, the alignment score of the "Beat" class at the validation phase is not a reliable proxy of its performance in the testing dataset, which suggests that the validation process is unreliable for the "Beat" class. However, for the other classes, we observe positive correlations between the alignment scores at the validation dataset and at the testing dataset, which suggests that the validation process is reliable.

In the ablation study (Experiment 3), which we do to find out which features are relevant to our gesture class prediction task, we replace some features with random values to observe how the model performance is being affected. We start by replacing the entire input with random values and use it on the trained model we use for Experiment 2 (Table 5.5, All features are randomized), we observe that all the alignment scores are lower than the alignment scores of the random output result (Table 5.2), except for "Beat" which is 0.040, which only marginally outperforms the random output. Subsequently, when we use the intensity alone (Table 5.5, Using intensity only), we find again that the model's alignment scores fail to outperform the random output result. This result, however, does not prove that intensity is unrelated to gesture timing. This result also does not prove that it is impossible to learn the gesture timing from the intensity. Our model simply happens to largely ignore the intensity feature, yet the model still can predict some classes (as shown in Experiment 2 results). Finally, in the sub-experiment where we only use the fundamental frequency (Table 5.5, Using F_0 only), the alignment scores are similar to

what we get when we use all prosody features (Table 5.2). This result suggests that the fundamental frequency is tied and is very pertinent to the gesture timing.

In Experiment 4 where the eyebrow movements are counted as "Beat", in the case when we consider upward eye movements, we find that the alignment score for the "Beat" class is slightly higher when we consider both the upward and the downward eyebrow movements (Table 5.6). However, more importantly, when we consider the eyebrow movements, the correlation between the alignment scores on the validation dataset and on the testing dataset becomes positive (Table 5.7), which suggests that the validation process becomes reliable. In fact, when the "Beat" includes both the upward and downward eyebrow movements, the correlation score becomes 0.925, which is close to one (i.e. a perfect correlation), which in turns suggests a highly reliable validation process. Besides that, the mean alignment scores on the validation dataset and on the testing dataset when we use both the upward and downward eyebrow movements (0.226 and 0.219 respectively) do not decrease much from when we only use the hand (0.202 and 0.2444 respectively), which suggests that the training reliability does not decrease.

In Experiment 5 where we use Mel-frequency cepstral coefficients instead of the prosody features (Table 5.8), we find that the alignment scores of "Beat", "IdeationalStroke", and "IdeationalOther" outperform the random output (Table 5.2). However, the alignment score of "IdeationalStroke" is considerably lower than the output from the model trained with prosody features (Experiment 2, Table 5.3). On the "Beat" class, the mean alignment scores on the validation dataset and on the testing dataset (Table 5.9) are 0.060 and 0.084 respectively, which are higher than the random output (0.009), which suggests that the training phase is still reliable. On the validation reliability, we however find that the correlation between the alignment scores on the validation dataset and on the testing dataset of the "Beat" class is only -0.056, which close to zero. However, the correlation scores are positive on all other classes. This is similar to what we find when we use prosody features (Experiment 2, Table 5.4), where the correlation score is close to zero on the "Beat" class but is positive on all other classes.

In Experiment 6 where we use both the prosody features and Mel-frequency cepstral coefficients (Table 5.10), the alignment score of "IdeationalOther" is 0.362, which is higher than we use prosody features only (Experiment 2, Table 5.3) but is the same when we use Mel-frequency cepstral coefficients only (Table 5.8). Meanwhile the alignment score of the "Beat" falls to 0. Besides that, the correlation between the alignment scores on the validation dataset and on the testing dataset on the "Beat" class is only 0.025 (Table 5.11), which is close to zero, which suggests that the validation process is unreliable for the "Beat" class. This unreliable validation process on the "Beat" class is similar to what we find in Experiments 2 and 5 (Tables 5.4 and 5.9). However, we find that the mean alignment scores for "Beat" on the validation dataset and on the testing dataset (Table 5.10) is also lower than when we use prosody features only (Experiment 2, Table 5.4) or when we use Mel-frequency cepstral coefficients only (Experiment 5, Table 5.9), which suggest that even in the training phase alone, using both prosody features and Mel-frequency cepstral coefficients lead to worse alignment score on the "Beat" class.

In Experiments 2, 5, and 6 we find that the mean alignment scores of the "Beat", "IdeationalStroke", and "IdeationalOther" (Tables 5.4, 5.9, and 5.11) are higher than the alignment scores of those classes in the random output (Table 5.2). These suggest that the training processes are reliable on those classes, no matter whether we use the three prosody features, the Mel-frequency cepstral coefficients, or both of them.

In Experiment 7 we train the model with one speaker and test it on the other speaker of the same interaction (Table 5.12). The idea behind this experiment is to find out whether a model trained on one speaker can be used with his/her conversation counterpart. We find that the alignment scores of the models outperform the random output (Table 5.2), which suggests that the model can indeed be used on the conversation counterpart.

5.6 Subjective Experiment

In the subjective experiment, 31 respondents watched 12 videos online of a virtual agent speaking and performing communicative gestures. Among the 31 respondents, 17 (55%) are male, 13 (42%) are female, and 1 (3%) refuses to disclose the gender. On the respondent breakdown by age, 6 (19%) are between 18-20 years old, 20 (65%) are between 21-30 years old, 2 (6%) are between 31-40 years old, and 3 (10%) are between 41-50 years old.

The 12 videos consist of 6 pairs. We extracted six speech segments from the Gest-IS corpus (see Chapter 4). We replicated the gestures of the human speaker on a virtual agent. Each pair of videos consists of the baseline and the output of the gesture generation model. The baseline videos have their gesture timings decided by randomly shuffling the gesture timing of the ground truth. In both baseline and model output videos, we retain the gesture shapes from the ground truth. In both videos, the virtual agents have the same appearance and speaking the same thing. We also use the original voice from the corpus. Therefore, the differences in the video pairs are only in the timing of the gestures. The animation of the agent contains only the arm gestures. There is no other animation (no head motion, gaze, posture shift, etc.). Moreover, we blur the face of the agent because having an agent with a still blank face might distract the respondents. The sequence of the 12 videos is shuffled to avoid having video pairs shown consecutively. A frame of a video is shown in Figure 5.5.

Our objective is to compare the respondent's perception differences between the videos based on the output of the gesture generation model and the baseline videos. We compare the naturalness, the time consistency, and the semantic consistency of the videos. For each of those dimensions, we measure it by asking the respondents to answer three questions. Each question asks the user to give a rating in likert scale from one to five. We sum the respondent's scores on the three questions to get the score of the dimension we want to measure. Therefore, for each dimension, the score ranges from 3 to 15 (inclusive of both values). The questions are listed in Table 5.13. They are adapted from the subjective study done by Kucherenko et al. (2019). We add one trick question for each video to ensure that the respondents were actually paying attention. We find that in all the three dimensions, namely naturalness, time consistency, and semantic consistency, the videos created based on the output of the gesture generation model have a higher average score. We also perform one-way ANOVA test to check the significance of the differences.

5.7 Subjective Experiment Results

In our subjective experiment, we measure the naturalness, the time consistency, and the semantic consistency of the gestures and speech. We compare the perception by human participants of the animation of the virtual agent where we manipulate the timing of the gestures. It allows us to measure the impact of the timings generated by the neural

Naturalness
How natural are the gestures? How smooth are the gestures? How appropriate are the gestures?
Time Consistency
How well does the gesture timing match the speech? How well does the gesture speed match the speech? How well does the gesture pace match the speech?
Semantic Consistency
How well do the gestures match the speech content? How well do the gestures describe the speech content? How much do the gestures help you understanding the speech content?

Table 5.13 – Subjective experiment questions (adapted from Kucherenko et al. (2019))

	Random Output Mean Score	Model Output Mean Score	p-value
Naturalness	8.565	9.796	1.04×10^{-5}
Time Consistency	8.565	10.409	7.271×10^{-9}
Semantic Consistency	7.855	9.457	4.487×10^{-6}

Table 5.14 – Subjective experiment results

Figure 5.5 – An example of a video frame in the subjective experiment

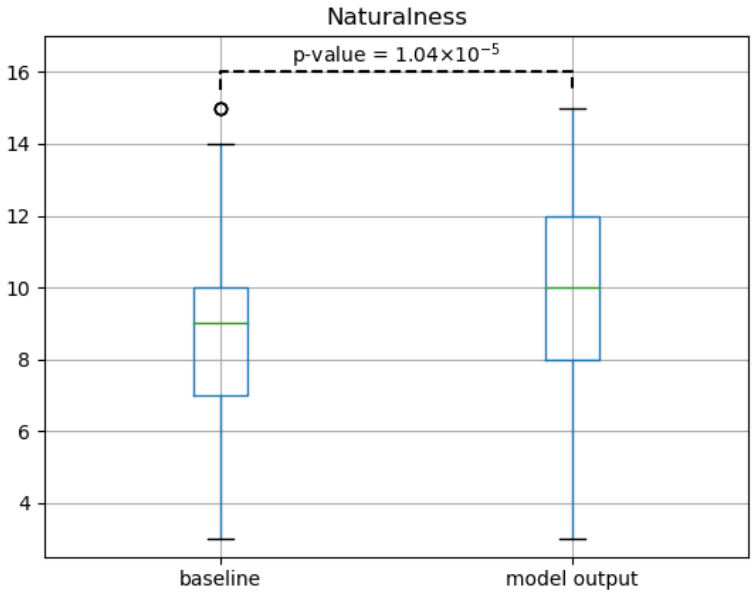

Figure 5.6 – The naturalness boxplot

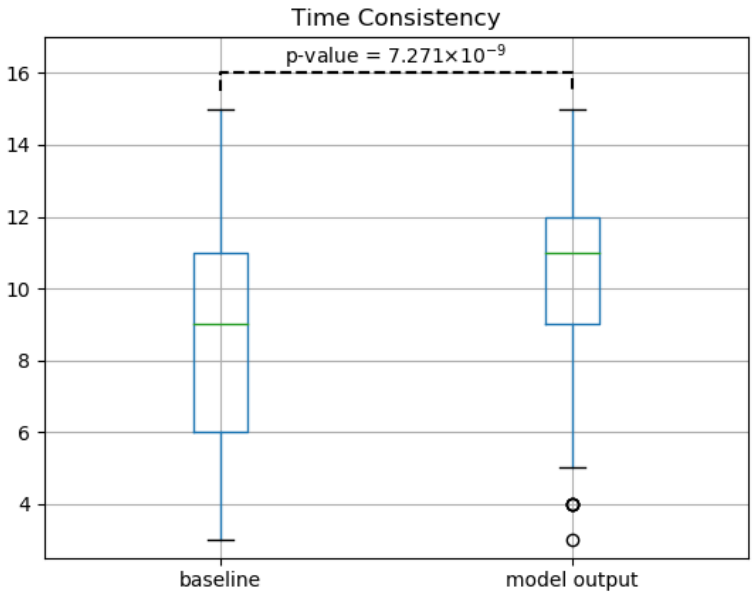

Figure 5.7 – The time-consistency boxplot

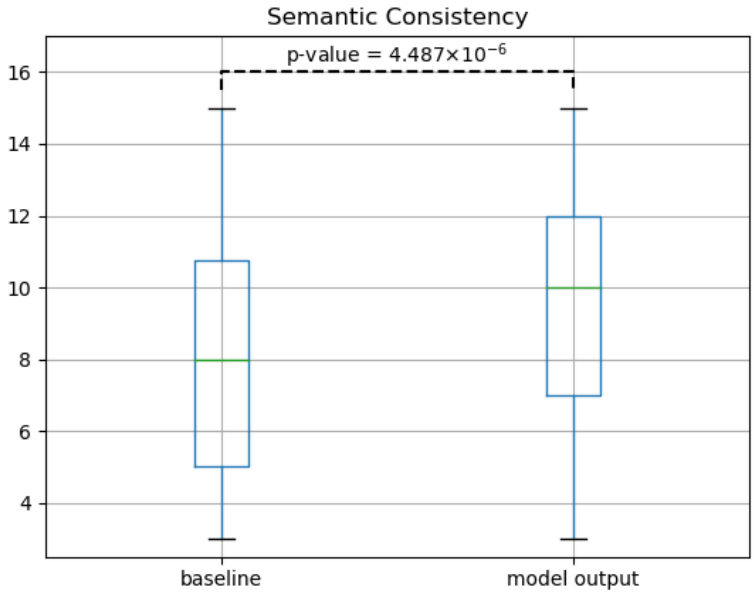

Figure 5.8 – The semantic-consistency boxplot

network model against random timings. The random timings act as the baselines. The idea is similar to what we do in our objective evaluation (Experiments 1 and 2, Tables 5.2 and 5.3). We find that the videos whose timing come from the model are rated better than the baseline in all the three measured qualities, and the differences are all significant ($p - value < 0.05$). The results are furnished in Table 5.14. We also show the boxplots of the three measured qualities (Figures 5.6, 5.7, 5.8).

5.8 Discussion

We observe in the performance of the random output ("Exp 1: Random output result", Table 5.2), different classes need classifier of different complexity. For example, the classifier of "Beat" (alignment score of 0.009) has to be more complex than the other classes are. On the other hand, the "NoGesture" class, with the alignment score of 0.533, can work with a classifier with a lower complexity, despite the fact that we select our samples only when the person is speaking. The result of Experiment 1 might be caused by the data imbalance. The "NoGesture" class is 50% larger than the "IdeationalStroke" class and is almost 300% larger than the "Beat" class. The rarity of "Beat" might cause the prediction to have a lower performance than it is for "NoGesture". Besides that, our corpus is small, with only 798 samples, which makes the training hard.

We observe that the performance of the model which is trained and tested with the entire data outperforms the random output's alignment scores on most classes. The validation process is not reliable on the "Beat" class, but the trained models still yield alignment scores higher than the random output, which suggests that the training process is still reliable. All these suggest that the three prosody features, namely fundamental frequency, fundamental frequency direction score, and intensity enable prediction of the gesture classes with a certain degree of reliability. However, as we have noted earlier

about Experiment 1, "Beat" is rare in our corpus, which makes the difficulty of predicting it expected. This leads us to the question on whether we would be able to predict "Beat" better if we have more data. Besides that, "Beat" gestures are not necessarily performed by hands; they can also be performed by head or facial movements (Bolinger (1989); Ekman (1979); Krahmer and Swerts (2004)). Indeed, in Experiment 4, we find that the validation reliability improves when we include the eyebrow movements (Table 5.7). These results are in line with the previous findings (Bolinger (1989); Ekman (1979); Krahmer and Swerts (2004)) which show that beat gestures can also be performed by eyebrow movements. It should also be noted that in our data, there are far more beat gestures done by eyebrow movements than by hands. There are around five times as many eyebrow beat time steps as hand beat time steps (1106 time steps by hand only vs 5679 time steps by eyebrow movements). Therefore, including eyebrow movements also increases the amount of "Beat" data.

On the "IdeationalStroke" class, our predictor is able to surpass the random output generator. This class encompasses the stroke of all communicative gestures except beat gestures. The model is able to predict where a gesture stroke (other than beat) is aligned with the acoustic features which we consider. This phase is well-studied in gesture literature as it carries the meaning of the gesture. This phase usually happens around or slightly before the pitch accent (Wagner et al. (2014)). In our case, we have the intensity, fundamental frequency, and fundamental frequency direction score features as our input. These three prosody features participate in the characterization of the pitch accent. We also find that our validation result is reliable, because the alignment scores on the validation dataset and on the testing data set show a positive correlation.

On the "IdeationalOther" class the model yields an alignment score higher than the random output, but the alignment score is still low. As a reminder, this class contains all the gesture phases (e.g., preparation, hold, retraction) except the stroke phase for all non-beat gestures. We can notice that, in all our experiments, we never obtain a good alignment on this class. This class is made of different gesture phases that may not correspond to the same prosodic profile. Their alignment may obey different synchronization needs (Wagner et al. (2014)).

In the ablation study (Experiment 3), we replace some features with random values to observe how the model performance is affected. We find that the alignment scores we get when we use the fundamental frequency (F_0) only are similar to what we get when we use all prosody features. This result suggests that the fundamental frequency is tied and is very pertinent to the gesture timing. This is in line with the findings that pitch accent and gesture stroke timing are related (Wagner et al. (2014)).

In Experiment 5 where we use Mel-frequency cepstral coefficients instead of the prosody features, we find that the performance is worse than when we use the model trained with prosody features. One possible reason is because the Mel-frequency cepstral coefficients are represented as a vector of 13 dimensions while the prosody features are represented as a vector of 3 dimensions. The higher dimension makes the search space much larger, and thus making the training slower as well. It should be noted that we train the models with "equivalent effort" (i.e., the same number of epochs and the same number of trainings). Besides that, the small corpus size is especially problematic if we want to learn a higher dimensional data. Another possible reason is that the Mel-frequency cepstral coefficients are indeed less informative about stroke timing than the prosody features. Indeed, it has been reported in several studies that F_0 are related to gesture stroke timing (Wagner et al. (2014)).

In Experiment 6 where we use both the prosody features and Mel-frequency cepstral coefficients, we find that the performance is also worse than when we use model trained with prosody features only. Although having both the Mel-frequency cepstral coefficients input and the prosody input enable the neural network to learn more information, it also makes the search space larger, which in turn makes the search slower.

In Experiment 2, 5 and 6, we observe that the validation reliabilities behave similarly: the correlation score is close to zero on "Beat" but positive on other classes. It suggests that simply replacing the input acoustic features does not fix the problem. However, in Experiment 4 when we include eyebrow movements as "Beat", we find that the validation reliability of the "Beat" improves. In Experiment 2, 5 and 6, we notice that the mean alignment scores on the validation dataset and on the testing dataset of the "Beat", "IdeationalStroke", and "IdeationalOther" exceeds the alignment scores at the random output. It suggests that the training processes are reliable on the "Beat", "IdeationalStroke', and "IdeationalOther" classes.

In Experiment 7 where we train the model with one speaker and test it on the other speaker of the same interaction, we find that the alignment scores of the models'outperform the random output, which suggests that some generalizability exists even-though people have different gesturing styles. These results may also be due to the fact that both speakers are parts of the same interaction and conversation participants tend to automatically align to each other; The alignment happens at various levels, such as phonology, syntax and semantics (Menenti et al. (2012)), as well as gesture types (Wessler and Hansen (2017)). These various alignments make the conversation itself successful (Garrod and Pickering (2009)).

In our subjective experiment, we measure the naturalness, time consistency, and the semantic consistency of the gestures and speech. We find that the timing from the model outperforms the baseline in all measured qualities, and the differences are significant ($p - value < 0.05$). It shows that overall the generated result is perceived better by the human respondents along the three qualities. It also shows that gesture timing is important to how well-perceived the gestures are by humans. We keep the gesture shapes from the ground truth in both the output of our model and the baseline, we act only on the timing of the gestures, yet the output of our model is perceived more favorably.

5.9 Conclusion

We develop a neural network model by using recurrent neural network with attention mechanism to predict gesture timing according to the acoustic input. We use three prosody features, namely fundamental frequency (F_0), F_0 direction score, and intensity. The model successfully predicts the gesture timing. However, on the beat gestures, our validation process is unreliable. This issue is solved when we count the eyebrow movements as another form beat gestures. We also do an ablation study where we replace some of the three features with random values to observe the impact on the model's performance. In the ablation study, we find that the fundamental frequency is pertinent to the gesture timing prediction. We also experiment with using the Mel-frequency cepstral coefficients (13 dimensions) instead of the aforementioned prosody features (3 dimensions) as the acoustic input and with both the Mel-frequency cepstral coefficients and the three aforementioned prosody features as the input. We find that we get the best performance when we use the aforementioned three prosody features only. It suggests that keeping the model as simple as possible is a good course of action, especially when we do not have much data. We also

try to train and validate on the data of one speaker and test it on his/her conversation's counterpart, and we find that it indeed possible. Finally, we do a subjective experiment where we compare the naturalness, the time consistency, and the semantic consistency of the gesture-and-speech videos whose gesture timing is generated by the model, and we compare them against the videos whose gesture timing is generated randomly. The videos whose timing comes from the output of the model is judged favorably, by the participants on the three measured parameters.

Chapter 6

Image Schema Computation and Embedding

In this chapter, we propose a method to provide a vector representation of image schema. The proposed method is based on word embedding techniques. Based on these vector representations, we also calculate the distances between those image schemas. These distances are proxy of similarities/differences between the different image schemas. Finally, based on those distances, we show the image schemas which are close/similar to each other.

6.1 Introduction

There are some studies which relate image schema to metaphoric gestures (see Section 2.6). However, those works are at "theoretical" level which concern themselves with the presence of the relationship: the image schema is already known, and then they investigate whether the image schema has any relationship with the metaphoric gestures. However, to actually apply this image schema notion in the gesture generation problem, we need a way to extract the image schemas from the text. Ravenet et al. (2018a,b) solve this problem by proposing a technique where they extract the image schemas by using WordNet senses. Another relevant trend is the use of word embedding techniques to represent text (see Section 3.4) in the general applied machine learning problems. Word embedding techniques have also been applied for gesture generation problems by using machine learning (see Section 3.4 and 3.6). An interesting following question is whether we can represent the image schema by using word embedding. However, word embedding also has a property such that two words with similar meanings are mapped to two vectors which are close to each other, which makes calculating the similarity/difference between the two words possible. This is the spirit behind the work in this chapter: we want to create vector representation of image schema, and with that we can quantify their similarities/differences.

In this chapter, we give a brief overview of WordNet in Section 6.2. Then, in the Related Work section, we explain the existing image schema computation technique by Ravenet et al. (2018a,b) and we also explain about word embedding. Then, we discuss about the limitations of the existing image schema computation technique and our proposal to improve the technique in Section 6.4. After that, in the Section 6.5, we explain

our method to get the embedding vectors for image schemas and how we can measure the distance/similarity between different image schemas. After that, in the Section 6.6, we show our experiments on obtaining the embedding vectors for the image schemas and our experiments pertaining to the distance/similarity between different image schemas. In the Section 6.7, we interpret the data from our experiments and what it implies for our image schemas. Finally, we conclude this chapter in Section 6.8.

6.2 Background

6.2.1 WordNet

WordNet (Miller (1995)) is a lexical database of English. It is organized as a directed graph. Each node is a "sense" (i.e. meaning), which is called "synset". Each sense has uniquely one part-of-speech tag (i.e. noun, verb, adjective, or adverb), but can have multiple lexemes associated with the sense. For example, a verb with the sense of being cognizant or aware of a fact or a specific piece of information can be represented with one of these lexemes: "know", "cognize", "cognise". WordNet only keeps the lexeme; therefore, the conjugation differences are eliminated. For example, "go", "goes", "went", and "gone" belong to the same lexeme. "Child" and "children" also belong to the same lexeme. Similarly, "high" and "higher" also belong to the same lexeme. However, different spellings of the same word are considered to be different lexemes. For example, "organize" and "organise" are different lexemes. A combination of a lexeme and a part-of-speech-tag can have multiple possible senses. For example, the noun "fan" has three possible senses: a device for creating a current of air by movement of a surface/surfaces, an enthusiastic devotee of sports, or an ardent follower and admirer. The senses are ordered from the most common sense to the least common sense. Two senses might have an edge connecting them. There are several edge types: "synonym", "antonym", "hypernym", "hyponym", "meronym" (one is a part of another), "troponym" (i.e. manner of doing), and "entailment". The WordNet's schema can be seen in Figure 6.1. Each WordNet sense belongs to uniquely one WordNet supersense, but one WordNet supersense can have many senses. The list of the supersenses is furnished in Table 6.1. It should be noted that supersense is a different property from sense. Sense is not related to supersense via the WordNet edges. We also furnish the statistics of the senses and the lexemes in WordNet at Tables 6.2 and 6.3. It can be seen in the Table 6.2 that noun senses far outnumber all other senses. Interestingly, there are also far more adjective senses than adverb senses, even though adjective and adverb are strongly related in English. Similarly, in Table 6.3, we observe again that the nouns far outnumber the rest. Similarly, the adjectives also outnumber the adverbs in terms of lexeme count.

6.3 Related Work

6.3.1 Existing Image Schema Computation

There are prior works about image schema and metaphoric gestures (see Section 2.6). However, those are concerned only on the theoretical underpinning, namely about presence of the relationship. On the other hand, to actually apply image schema for gesture generation problem, we need a way to infer the image schema from a free-form text (i.e.

Adjective	Adverb	Noun	Verb
all	all	act	body
pert		animal	change
ppl		artifact	cognition
		attribute	communication
		body	competition
		cognition	consumption
		communication	contact
		event	creation
		feeling	emotion
		food	Framestext
		group	motion
		location	perception
		motive	possession
		object	social
		person	stative
		phenomenon	weather
		plant	
		possession	
		process	
		quantity	
		relation	
		shape	
		state	
		substance	
		time	
		Tops	

Table 6.1 – The WordNet's supersenses

Type	Count
Noun	82115
Verb	13767
Adjective	18156
Adverb	3621

Table 6.2 – WordNet 3.0's sense count

Type	Count
Noun	117798
Verb	11529
Adjective	21479
Adverb	4481
All	147306

Table 6.3 – WordNet 3.0's lexeme count. A lexeme may cover several part-of-speech types

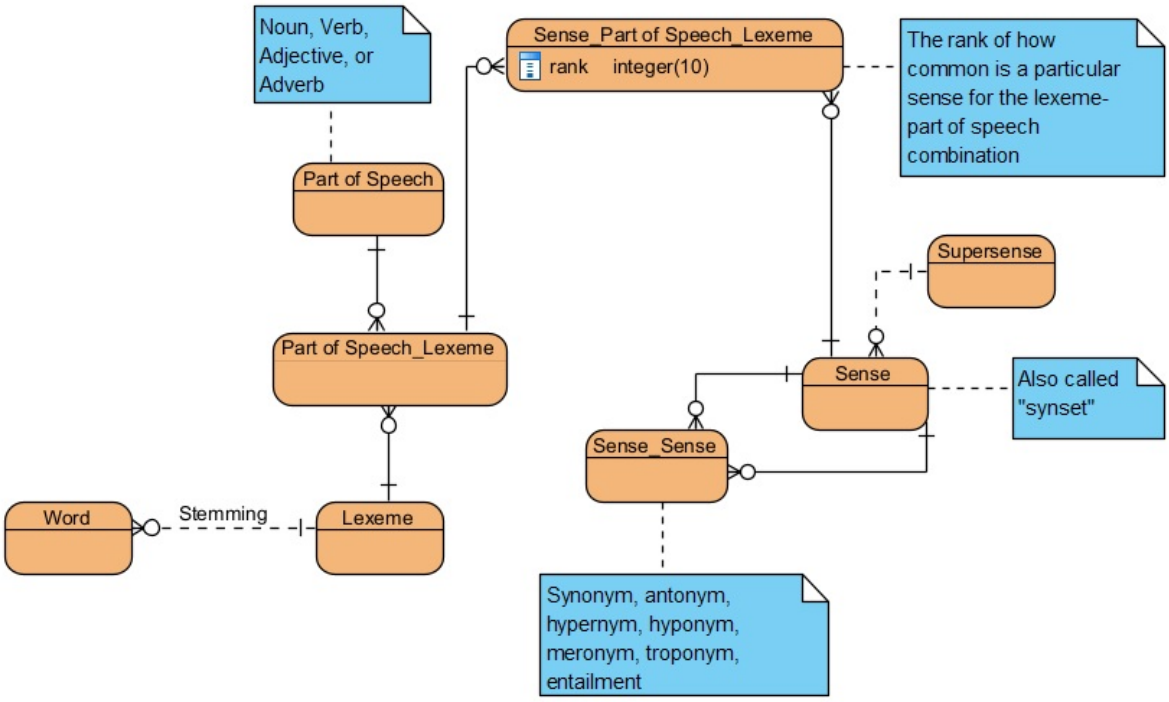

Figure 6.1 – WordNet's schema in Entity-Relationship diagram

the content of the speech). Ravenet et al. (2018a,b) develop a technique to solve this problem.

The technique of Ravenet et al works by first parsing the text to get the corresponding parts of speech. This parsing is done by using the Stanford POS Tagger (Toutanova et al. (2003)). Then, the technique gets the lexeme of the word. For example, “walk”, “walks”, and “walked” will be converted into “walk”. Similarly, both “hand” and “hands” will be converted into “hand”. It should be noted, however, gendered nouns exist but are rare in English (e.g. “actor”-“actress”, “governor”-“governess”). These gendered nouns are considered as different words. Then, based on the lexemes of the word (e.g. “walk” for “walks”) and the part of speech (e.g. verb), the technique decides the correct sense by using simplified Lesk algorithm (Lesk (1986)). The simplied Lesk agorithm works by calculating the size of the intersection between the possible lexemes of each sense and the lexemes in the rest of the sentence. The sense whose size of the intersection is the largest is considered to be the correct sense. Once the correct sense is decided, then the WordNet sense graph is traversed through the hypernym edges until it finds one of the “terminal senses/nodes”. The traversal is done by a depth first search traversal. Ravenet et al's algorithm has a mapping set by static rules which map a set of senses/nodes to an image schema. Thus, the relationship between the image schema and the terminal senses/nodes is one-to-many. The image schema is decided based on this terminal WordNet sense. If there are multiple words with image schemas in a phrase, then we prioritize the word which is tagged to have pitch accent. The pitch accent is relevant because the stroke often happens at around the pitch accent (Kendon (2004)), and thus the gesture is more likely to depict the word where the pitch accent is. Ravenet et al's algorithm is furnished in Algorithm 1.

```
split the sentence into phrases and the POS tags;
for phrase in all phrases do
    for word and POS tag in the phrase do
        if has pitch accent or is adjective or is adverb then
            mark this word as priority
        end
        lexemes ← getPossibleLexemes(word, POS_tag);
        for lexeme in lexemes do
            mostLikelySense ← simplifiedLeskAlgorithm(lexeme, POS_tag);
            //traverse the WordNet graph via hypernym edges;
             imageSchemas+ =
             getImageSchemaByTraversingWordNetGraph(mostLikelySense);
        end
        if has image schema then
            chosenImageSchema = imageSchemas[0]
        end
    end
    for word and POS tag in the phrase do
        if priority word has image schema then
            choose this image schema;
        else if non priority word has image schema then
            choose this image schema;
        else
            this phrase has no image schema;
    end
end
```

Algorithm 1: Ravenet et al. (2018a,b)'s algorithm to compute image schema

6.3.2 Word Embedding

One recent trend in natural language processing is the use of word embedding to represent a word in the form of a vector. This vector representation can then be used in various machine learning models for various problems. For example, when BERT (Devlin et al. (2018)) is being proposed, it is tested on natural language understanding tasks, questions answering tasks, and sentence continuation tasks.

Word embedding has a property that two words which have similar meanings have their vector representations also close to each other. It should be noted, however, word embedding is not a single technique, instead it is better seen as a class of techniques which shares the same principle, but is implemented differently. Especially, different techniques have different notions of similarity and how this similarity is translated into distance in the vector space. The distance between the embedding vectors are calculated by using the euclidean distance or the cosine distance.

Word2Vec (Mikolov et al. (2013)) is the pioneer of word embedding. It uses the notion that two words are similar if they are surrounded by the same words. There are two variants of Word2Vec. The first one is basically a neural network which does a "fill in the blank" task. Given an n-gram whose middle word is hidden, the network is to guess the hidden word. This is called "Continous Bag Of Words" (CBOW) approach. The second variant is the reverse: given the middle word, the network is to guess the rest of the n-gram. This is called "Skip-Gram" approach. The result of any of these approaches is that, two words which are surrounded by similar words (in an n-gram) will have the corresponding vectors also close to each other. Ahuja and Morency (2019) use Word2Vec to represent the text in their machine learning system to predict body pose according to the text input. Pouw et al. (2021) compare the Word2Vec cosine distance of the text and the kinematic distance of the head tip, wrist, and head movement. The kinematic distance is calculated by using Dynamic Time Warping. Interestingly, Pouw et al find a weak but reliable relation between the Word2Vec distance and the kinematic distance.

GloVe (Pennington et al. (2014)) follows a similar notion. However, While Word2Vec treats different n-grams as different samples, GloVe learns from the global data of co-occurences. GloVe works by doing an optimization such that the distance between the two corresponding vectors are minimized when their probabilities of co-occurence with other words are similar. Effectively, it means two words are similar if they tend to co-occur with similar words.

Both Word2Vec and Glove have "static" embedding. A word always has the same embedding, no matter the context where it appears. However, one word can have different meanings depending on the context. For example, "fan" can mean a follower or admirer (e.g. "she is a fan of Justin Bieber") or an instrument to displace air (e.g. "I need a fan because my room is hot"). BERT (Devlin et al. (2018)) on the other hand, yields different embeddings for the same word depending on the context. This is called "contextual word embedding". BERT training works in two ways. The first one is that it takes the sentence as the input, with a few words being hidden, and then the network learns to reproduce the same sentence including the hidden words. Here, the network learns to "fill in the blank" (see Figure 6.2). The second one is that the network is given two sentences (some words are hidden), and the network learns to re-produce the complete sentences and to indicate whether the two sentences are consecutive. Here, the network learns if the context of the two sentences are related. Thus, it can be seen that unlike Word2Vec or GloVe which learn the word embedding in isolation, BERT learns the context together. Consequently, the same word will have different corresponding vectors depending on the

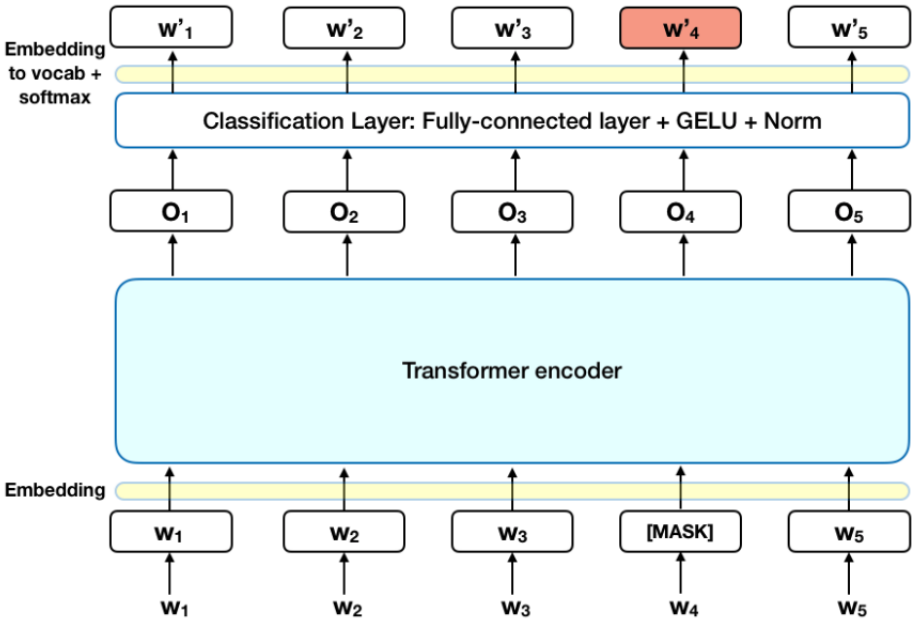

Figure 6.2 – The "fill in the blank" training of BERT (Devlin et al. (2018)) [a]. The network learns to predict w_4.

[a] The schema is from https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b2

sentence. Kucherenko et al. (2020) use BERT to represent the text in their machine learning system to generate body movements according to both the text and the acoustic of the speech (see Section 3.6).

SenseBERT (Levine et al. (2020)) is an extension BERT by also taking into account the similarities in WordNet in the training. Therefore, unlike BERT, SenseBERT explicitly works at the word sense level. Specifically, unlike BERT which learns the words only, SenseBERT learns both the words and the corresponding supersenses. Therefore, it can be said that SenseBERT consider two words to be more similar when they have the same corresponding supersense. Levine et al show that SenseBERT outperforms BERT on both SemEval word sense disambiguation tasks and "word in context" tasks. The schema depicting the difference between BERT and SenseBERT is available in Figure 6.3.

We can see from these works that word embedding can be used to represent text and there are many variants of word embedding. Therefore, we have three research questions we aim to address. The first one is how can we use word embedding to represent an image schema as a vector. The second one is how do we compare the word embedding we choose against the alternatives. However, if we have the image schema vectors, then the distances between different image schemas also become computable, and thus it makes sense to ask if certain image schemas are closer to each other. Therefore, in the third research question, we want to investigate which image schemas are close to each other.

6.4 The Limitations of the Ravenet et al's Algorithm

In this section, we discuss the limitations of Ravenet et al's algorithm and our proposal on how to address them. The first one is the limitation of the Lesk algorithm which Ravenet

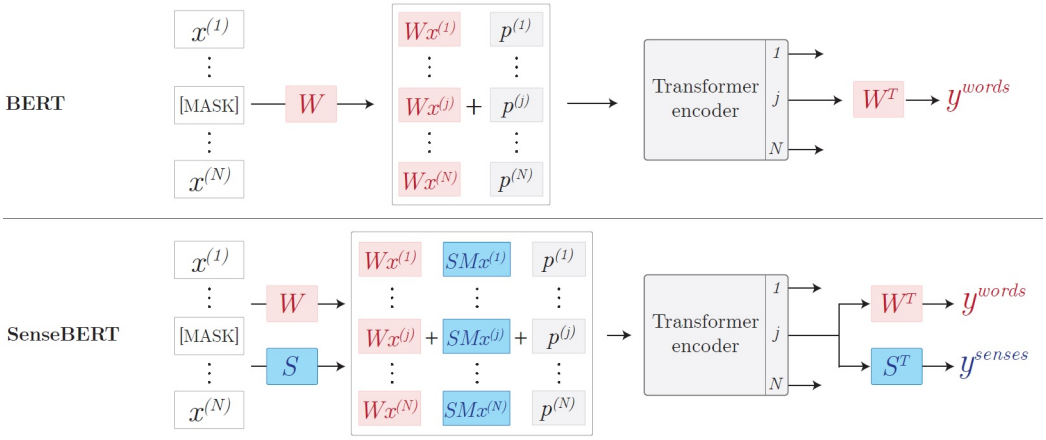

Figure 6.3 – The schemas of the difference between BERT and SenseBERT (Levine et al. (2020)). Unlike BERT, during the "fill in the blank" of SenseBERT, it tries to predict the WordNet supersense as well (see Table 6.1).

et al use in their algorithm (see Algorithm 1 Line 9) to disambiguate the word. In Section 6.4.1, we explain that some researchers compare the Lesk algorithm against other word sense disambiguation techniques and find that a far simpler algorithm, namely always choosing the WodNet's first sense, outperforms Lesk algorithm in several different experiments. The second one is the limitation of the hypernym-only WordNet graph traversal (see Algorithm 1 Line 10). In Section 6.4.2, we explain why using only the hypernym edges is not sufficient and how we propose to address this issue.

6.4.1 Limitation of Lesk Algorithm For Word Sense Disambiguation

Ravenet et al. (2018a,b) use Lesk algorithm for word sense disambiguation in their image schema computation technique. However, Lesk algorithm is not the only word sense disambiguation techniques. Raganato et al. (2017) compare many such techniques. The techniques include those which are based on machine learning, namely IMS (Zhong and Ng (2010)), IMS+embeddings (Taghipour and Ng (2015b); Rothe and Schütze (2015); Iacobacci et al. (2016)), Context2Vec (Melamud et al. (2016)), and also knowledge-based techniques, namely Lesk (Lesk (1986)), UKB (Agirre and Soroa (2009); Agirre et al. (2014)), and Babelfly (Moro et al. (2014)). The machine-learning based techniques are trained on SemCor (Miller et al. (1994)) and OMSTI (Taghipour and Ng (2015a)) corpora. Raganato et al also add two simple baseline techniques, namely Most Frequent Sense (MFS) technique which always chooses the most common sense in the training corpus and WordNet First Sense technique which always chooses the first WordNet sense. It should be remembered that the first WordNet sense is the most common sense. Then, Raganato et al compare those techniques in a uniform setting. They use the data from Senseval (Edmonds and Cotton (2001); Snyder and Palmer (2004)) and SemEval (Pradhan et al. (2007); Navigli et al. (2013); Moro and Navigli (2015)) corpora for testing. All the data have the part-of-speech tagging done by using Stanford CoreNLP toolkit (Manning et al. (2014)). All the corpora include WordNet sense annotations. They use F-Measure to quantify the performance of the Word Sense Disambiguation techniques. Interestingly, as can be seen in Table 6.4, the WordNet first sense always beats or close to beat the F-Score of all the knowledge-based techniques. In all cases, the WordNet first sense beats the

Corpus	Range of Machine Learning Techniques	Range of Knowledge Based Techniques	Lesk Algorithm	MFS	WordNet 1st Sense
Senseval-2	70.8 - 73.3	50.6 - 67.0	50.6	66.5	66.8
Senseval-3	68.2 - 69.6	44.5 - 63.7	44.5	60.4	66.2
SemEval-07	58.5 - 61.5	32.0 - 56.7	32.0	52.3	55.2
SemEval-13	65.0 - 67.2	53.6 - 66.4	53.6	62.6	63.0
SemEval-15	64.2 - 71.7	51.0 - 70.3	51.0	64.2	67.8

Table 6.4 – F-score comparison of the word sense disambiguation techniques (Raganato et al. (2017))

Lesk algorithm. Although WordNet first Sense is less powerful than the machine learning techniques, the WordNet first Sense stands out for its remarkable "cost-effectiveness". It is also remarkably effective compared to the knowledge-based techniques despite being extremely simple. Therefore, we propose to replace the use of Lesk algorithm with the far simpler method of always using the WordNet's first sense.

6.4.2 Limitation of Hypernym-Only WordNet Graph Traversal

As can be seen in Algorithm 1 Line 10, the WordNet graph traversal in the original Ravenet et al's algorithm is only done through the hypernym edges. However, WordNet has hypernym edge for noun and verb senses only (Miller (1995)). Therefore, we need a new method to traverse the WordNet graph. For adjective, we use the synonym relationship instead (e.g. "essential" is a synonym of "important"). It should be noted, however, synonym edge is bidirectional. x is a synonym of y if and only if y is a synonym of x. For adverb, we get the corresponding adjective by using the "derived from adjective" edge (e.g. "importantly" is derived from the adjective "important"), then we do as the aforementioned treatment of adjectives. For verb, we also use the troponym edge. Troponym itself means a manner of doing an action. For example, both "walk" and "fly" are troponyms of "move". It should be noted that for verb, hypernym and troponym are the inverse. x is a hypernym of y if and only if y is a troponym of x. The algorithms are furnished in Algorithm 2. We also present the statistics of about their connections in Table 6.5. It can be seen from the number of the connections and the number of senses/nodes for each part of speech (Table 6.2) that the graph is very sparse: there are even less edges than nodes. It means, from any given node, there are only a few other nodes we can reach.

6.5 Proposed Method

One thing we can easily see is that there are far more English words than there are image schemas. For example, Ravenet et al. (2018a,b) list only 25 image schemas. Therefore, many different words are necessarily mapped to the same image schema. Considering that those which are mapped to the same image schema should have similar meanings, then the corresponding vectors should also be close to each other, and thus they should form a cluster. Therefore, to answer our first research question (i.e. how can we use word embedding to represent image schema as vector?), we will use the centroid of the word embedding vectors which belong to the same image schema as the embedding of the image schema.

```
1  function getAdjectiveImageSchema(Synset sense) {;
2      breadthFirstSearch(sense, allWordnetSenses, allSynonymEdges);
3  }
4  function getAdverbImageSchema(Synset sense) {;
5      adjectiveSense ←
       traverseOneHop(sense, allWordnetSenses, allDerivedFromAdjectiveEdges);
6      returngetAdjectiveImageSchema(adjectiveSense);
7  }
8  function getVerbImageSchema(Synset sense) {;
9      breadthFirstSearch(sense, allWordnetSenses, allHypernymEdges +
       allTroponymEdges);
10 }
11 function getNounImageSchema(Synset sense) {;
12     breadthFirstSearch(sense, allWordnetSenses, allHypernymEdges);
13 }
```

Algorithm 2: The proposed computation of image schema

Type	Count
Between nouns (hypernym)	75850
Between verbs (hypernym & troponym)	13238
Between adjectives (synonym)	10693

Table 6.5 – WordNet 3.0's unordered connected sense count

There are multiple word embedding techniques which can map a word into a vector (see Section 6.3.2). BERT is a popular word embedding for a general purpose natural language processing application and has been used in gesture generation problem as well (Kucherenko et al. (2020), see Section 3.6), and thus BERT is a reasonable choice. However, considering that Ravenet et al. (2018a,b) use WordNet to map the words into image schemas, a more suitable word embedding might be one which also takes into account the WordNet similarity. SenseBERT is one such embedding technique. Therefore, we will try both BERT and SenseBERT and we will compare their results.

The image schema tagging in Ravenet et al's algorithm is done per phrase (see Algorithm 1 Line 2). A sentence is split into several phrases (noun phrase, verb phrase, etc.), and then there is at most one image schema per phrase. However, we know from which word the image schema comes from (see Algorithm 1 Line 8). Therefore, we try two variations of the way to get the vector: by averaging all words in the phrase, or by getting the vector of the word where the image schema comes from.

There are two possible distance metrics to measure distance between two embedding vectors, namely euclidean distance and cosine distance. Both distance metrics are often used to measure distance between word embedding vectors. We will try both distance metrics to find out if they yield different results.

Having two different word embedding techniques as the possible choices leads us to our second research question: how do we compare the word embedding we choose against the alternative? To answer this question, we will find out which word embedding method gives a better clustering behavior. For that, we use two measures. The first one is cluster purity and the second one is comparing the intra-cluster against the inter-cluster distances.

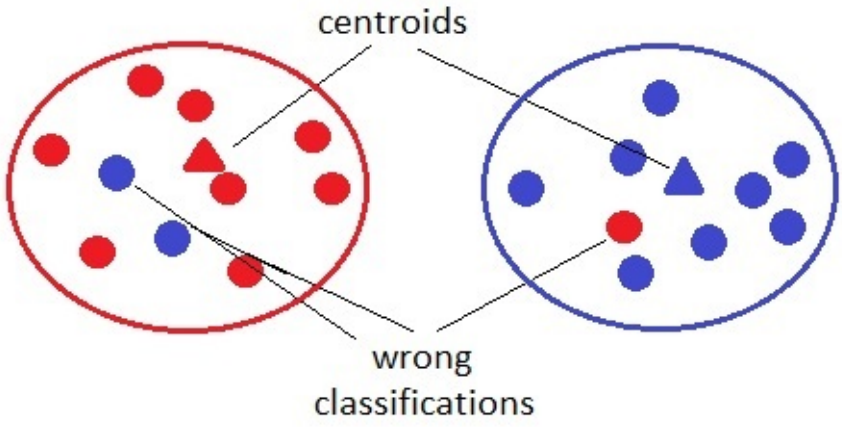

Figure 6.4 – An illustration of the notion of cluster purity. The different colors (i.e. red and blue) represent different image schemas, the small solid circles represent vectors, the triangles represent the centroids, and the large hollow circles represent classifications. The three wrongly classified vectors are misclassified because they are closer to another centroid than to their own centroid. The cluster is purer if there are less misclassified vectors.

A better clustering behavior yields purer clusters. That means, vectors which belong to a certain image schema should be closer to the centroid of that image schema than to any other centroids. Equivalently, vectors which do not belong to a certain image schema should not have their closest centroid to be the centroid of that image schema. For that, we do a "classification" by using the nearest cluster centroid. We "classify" each vector according to the nearest centroid. If the nearest centroid is indeed the centroid of its image schema, then we consider it as a correct classification, or else it is a wrong classification. The illustration of this cluster purity notion is furnished in Figure 6.4. According to the results of these classifications, we measure the F_1 score of each class. This calculation is done in one-vs-rest manner. That means, when we calculate the F_1 score of "OBJECT" image schema, we do a binary classification of "OBJECT" against all other image schemas. After that, we get the multi-class F_1 score from the weighted average of the F_1 scores of each class. The weight is proportional to the number of vectors in that image schema/-cluster. This weighting is to take into account the fact that some image schemas are more frequent than the others. In this measurement, a higher F_1 score is better. The formula is shown in Formula 6.1. In that formula, TP stands for the number of "True Positive"s, FP stands for the number of "False Positive"s, and FN stands for the number of "False Negative"s of the aforementioned classification. The $\frac{|v \in IS|}{|v|}$ multiplier in the second line of the formula is the weight of the class's F_1 score.

The second measure is to compare the intra-cluster against the inter-cluster distances. Basically, the distances within a cluster should be closer than the distance beyond one cluster. For this, we calculate the inertia score (Formula 6.2). Specifically, we compare the distance between the centroids of each image schema to the global centroid (i.e. the center of all vectors) against the distance between each data point to its cluster centroid. If the distance between each cluster centroid to the global centroid is large and the distance between the individual vectors to their respective centroid is small, then the inertia score will be high. The illustration of this metric is provided in Figure 6.5. In our measure, a

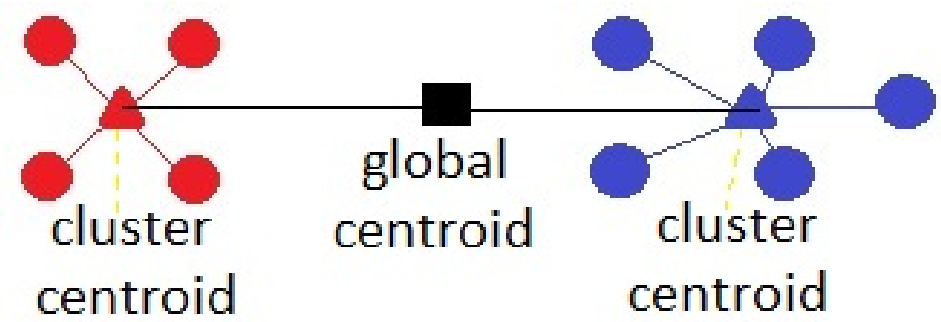

Figure 6.5 – An illustration of the notion of cluster inertia. The different colors (i.e. red and blue) represent different image schemas, the circles represent vectors, the triangles represent the cluster centroids, the black rectangle represents the global centroid, the red or blue lines represent the intra-cluster distances, and the black lines represent the inter-cluster distances. The inertia score is higher if the black lines (i.e. the intra-cluster distances) are longer than the red or blue lines (i.e. the intra-cluster distances)

higher inertia score signifies better clustering. We weight each clusters/image schemas linearly to the number of vectors in that image schema to take into account the fact that some image schemas are more frequent than the others.

$$\begin{aligned} F_{1_{IS}} &= \frac{TP}{TP + \frac{FP+FN}{2}} \\ F_1 &= \sum_{IS} \frac{|v \in IS|}{|v|} F_{1_{IS}} \end{aligned} \tag{6.1}$$

$$inertia = \sum_{IS} \frac{|v \in IS| \times d(c_{IS}, c_{global})}{\sum_{v \in IS} d(v, c_{IS})} \tag{6.2}$$

Once each image schema can be represented as a vector, then the notion of distance becomes sensible. Previously, we have a notion of different image schemas, but we cannot tell if some image schemas are closer or more similar to each other than to the others. However, if each image schema is represented by a vector, then we can calculate the distance between them. This is to answer our third research question: which image schemas are close to each other? We will measure their distances and also do a hierarchical clustering to show how similar image schemas can be merged. We use two measures to calculate the distance between different image schemas.

In the first measure (Formula 6.3), we calculate the distance between their respective centroids. If the two centroids are near, then the two image schemas are considered to be similar.

In the second measure, we measure the confusion between the individual vectors of each image schema (Formula 6.4). Specifically, given two image schemas IS_1 and IS_2, we measure how many individual vectors belonging to image schema IS_1 which are closer to the centroid of IS_2 and vice versa. The more numerous such vectors are (i.e. the more "confused" the vectors are), the closer the two image schemas are. This is essentially the inverse of the cluster purity measure. We normalize the number of the "confused" vectors against the size of each image schema. Therefore, this confusion distance works at the level of proportion. For example, if only 30% of IS_1's vectors are closer to the centroid of

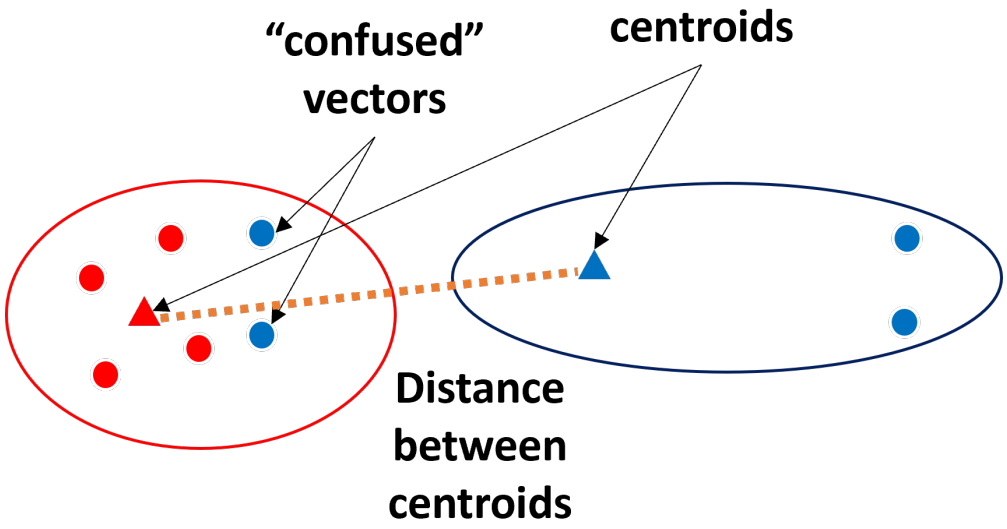

Figure 6.6 – An illustration of image schema distance. The different colors (i.e. red and blue) represent different image schemas, the circles represent vectors, the triangles represent the cluster centroids, and the orange dotted line represents the distance between the two centroids. Two blue vectors are nearer to the red centroid, and thus they are “confused”. On the distance metric which uses the inter-centroid distance (Formula 6.3), the two image schemas are closer to each other if the dotted orange line is shorter. On the distance metric which uses the confusion (Formula 6.4), the two image schemas are closer to each other if there are more “confused” vectors.

IS_1 (than to the centroid of IS_2) and only 40% of IS_2's vectors are closer to the centroid of IS_2 (than to the centroid of IS_1), then the confusion is $(0.3 + 0.4)/2 = 0.35$. The lower the score, the closer the two image schemas are.

Both of those distance metrics are illustrated in Figure 6.6. The confusion-based metric is essentially the inverse of the cluster purity measure. The main difference between those two metrics is that the confusion distance takes into account the spread of the vectors. For example, in Figure 6.6, the red cluster is more compact than the blue cluster. Two of the blue vectors are actually closer to the red centroid than to the blue centroid.

$$d_{centroid}(IS_1, IS_2) = d(c_{IS_1}, c_{IS_2}) \tag{6.3}$$

$$d_{confusion}(IS_1, IS_2) = \frac{1}{2}\left(\frac{|v \mid v \in IS_1 \wedge d(v, c_{IS_1} > d(v, c_{IS_2}))|}{|v \in IS_1|} + \frac{|v \mid v \in IS_2 \wedge d(v, c_{IS_1} < d(v, c_{IS_2}))|}{|v \in IS_2|}\right) \tag{6.4}$$

6.6 Experiment

We use the Stanford Natural Language Inference (SNLI) (Bowman et al. (2015)) corpus as the source of our sentences. This corpus has 652,505 sentences in English. First, we run Ravenet et al's algorithm to extract the image schema from those sentences and we mark the word where the image schema comes from. The image schema statistics is furnished

Image Schema	Count	Proportion
ATTRACTION	1368	0.0997%
BACK	2104	0.153%
BIG	13863	1.010%
CONTAINER	11147	0.812%
CONTRAST	234	0.017%
DOWN	3284	0.239%
EMPTY	1659	0.121%
ENABLEMENT	7980	0.581%
FAR	8830	0.643%
FRONT	1459	0.106%
FULL	1011	0.074%
IN	143151	10.431%
INTERVAL	69848	5.090%
ITERATION	406	0.030%
LEFT	1534	0.112%
LINK	19387	1.413%
MERGING	6146	0.448%
NEAR	10203	0.743%
OBJECT	550542	40.116%
OUT	13594	0.991%
RIGHT	8469	0.617%
SMALL	3087	0.225%
SURFACE	5932	0.432%
UP	25226	1.838%
WHOLE	461918	33.658%

Table 6.6 – Image schema count and their proportion

at Table 6.6. After that, we run both BERT and SenseBERT to get the embedding of each word.

In our first experiment, we measure the clusters'F_1 score/purity (see Formula 6.1) and inertia (see Formula 6.2) to compare BERT and SenseBERT on the resulting clustering behavior. We permute through the options of cosine distance - euclidean distance and embedding comes from the word - embedding is averaged through the phrase.

We furnish in Table 6.7 the inertia when we calculate the value of each vector as the average BERT/SenseBERT embedding vector of all the words in the phrase. On the other hand, in Table 6.8, we show the corresponding values if each vector is the BERT/SenseBERT embedding vector of the word where the image schema comes from.

Similarly, we furnish in Tables 6.9 and 6.10 the F_1 score when we calculate the value of each vector as the average BERT/SenseBERT embedding vector of all the words in the phrase and when the value if each vector is the BERT/SenseBERT embedding vector of the word where the image schema comes from.

In our second experiment, we want to find out which image schemas are close to each other. However, we have already represented the image schemas as clusters. Therefore, we measure the distances between their respective clusters as a proxy of similarities/differences between the different image schemas. For this, we measure the inter-centroid

	By Euclidean Distance	By Cosine Distance
BERT	0.047	0.072
SenseBERT	0.052	0.104

Table 6.7 – Inertia measure with each vector is calculated by averaging from all words in the phrase

	By Euclidean Distance	By Cosine Distance
BERT	0.048	0.082
SenseBERT	0.078	0.158

Table 6.8 – Inertia measure with each vector comes from the word where the image schema comes from

	By Euclidean Distance	By Cosine Distance
BERT	0.439	0.526
SenseBERT	0.613	0.609

Table 6.9 – F_1 score with each vector is calculated by averaging from all words in the phrase

	By Euclidean Distance	By Cosine Distance
BERT	0.627	0.619
SenseBERT	0.705	0.697

Table 6.10 – F_1 score with each vector comes from the word where the image schema comes from

Closeness Rank	BERT / Euclidean Distance	BERT / Cosine Distance
1	OBJECT-WHOLE	OBJECT-WHOLE
2	LINK-OBJECT	LINK-OBJECT
3	LINK-WHOLE	LINK-WHOLE
4	ENABLEMENT-OBJECT	ENABLEMENT-OBJECT
5	OBJECT-SURFACE	OBJECT-SURFACE

Table 6.11 – Five closest pairs of image schemas in BERT with the centroid distance

Closeness Rank	SenseBERT / Euclidean Distance	SenseBERT / Cosine Distance
1	OBJECT-WHOLE	OBJECT-WHOLE
2	LINK-OBJECT	LINK-OBJECT
3	ENABLEMENT-OBJECT	LINK-WHOLE
4	LINK-WHOLE	ENABLEMENT-OBJECT
5	OBJECT-SURFACE	OBJECT-SURFACE

Table 6.12 – Five closest pairs of image schemas in SenseBERT with the centroid distance

distances (see Formula 6.3) and the confusion rates (see Formula 6.4). It is useful to remember that the main difference between the inter-centroid distance metric and the confusion metric is that the confusion metric takes into account the spread of the cluster while the inter-centroid distance metric does not. We permute through the options of BERT - SenseBERT and cosine distance - euclidean distance. We furnish the five closest pairs of image schemas according to those parameters in Tables 6.11, 6.12, 6.13, and 6.14. We also show the hierarchical clusters of the image schemas according to the same permutation of parameters. In this hierarchical clustering, we merge the closest pair of image schemas at each step until there is only one remaining. The hierarchical clusters are displayed in Figures 6.7, 6.8, 6.9, 6.10, 6.11, 6.12, 6.13, and 6.14

We also display the UMAP visualizations (McInnes et al. (2018)) of three cases, namely EMPTY image schema in BERT and euclidean distance (Figure 6.15), RIGHT image schema in SenseBERT and cosine distance (Figure 6.16), and NEAR image schema in SenseBERT and euclidean distance (Figure 6.17). We choose them because according to the hierarchical clusters when we use the centroid distance, they are among the last to be merged,

Closeness Rank	BERT / Euclidean Distance	BERT / Cosine Distance
1	OBJECT-WHOLE	OBJECT-WHOLE
2	INTERVAL-UP	LINK-OBJECT
3	LINK-OBJECT	INTERVAL-UP
4	LINK-WHOLE	LINK-WHOLE
5	ENABLEMENT-OBJECT	BIG-UP

Table 6.13 – Five closest pairs of image schemas in BERT with the confusion distance

Closeness Rank	SenseBERT / Euclidean Distance	SenseBERT / Cosine Distance
1	OBJECT-WHOLE	OBJECT-WHOLE
2	INTERVAL-UP	INTERVAL-UP
3	BIG-DOWN	BIG-DOWN
4	LINK-OBJECT	LINK-OBJECT
5	BIG-UP	BIG-UP

Table 6.14 – Five closest pairs of image schemas in SenseBERT with the confusion distance

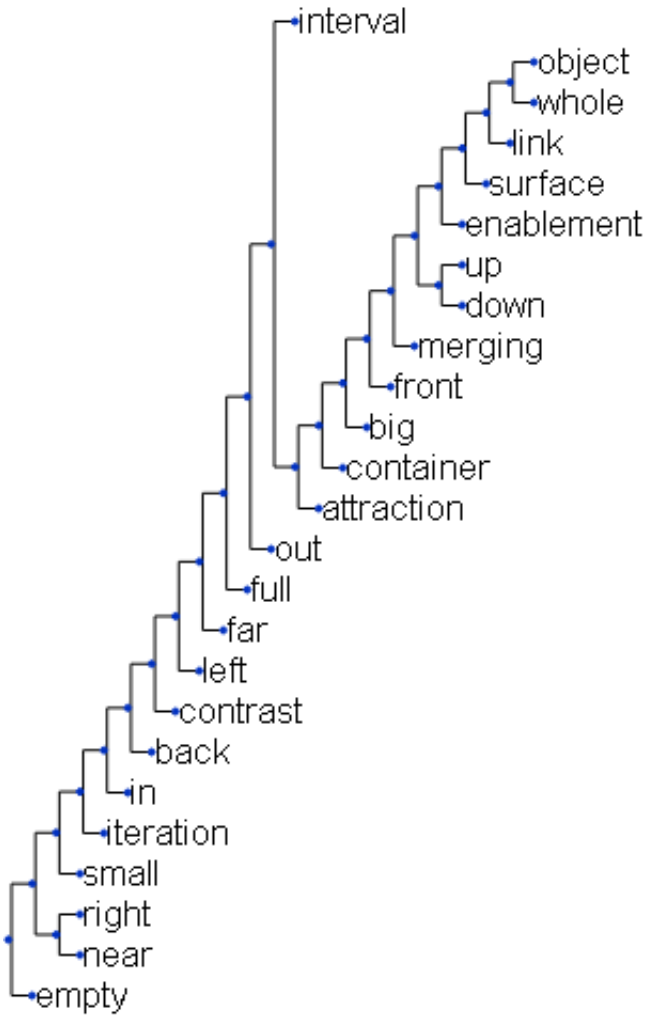

Figure 6.7 – The hierarchical clustering of image schemas in BERT and euclidean distance (between two vectors) with the centroid distance (between two image schemas)

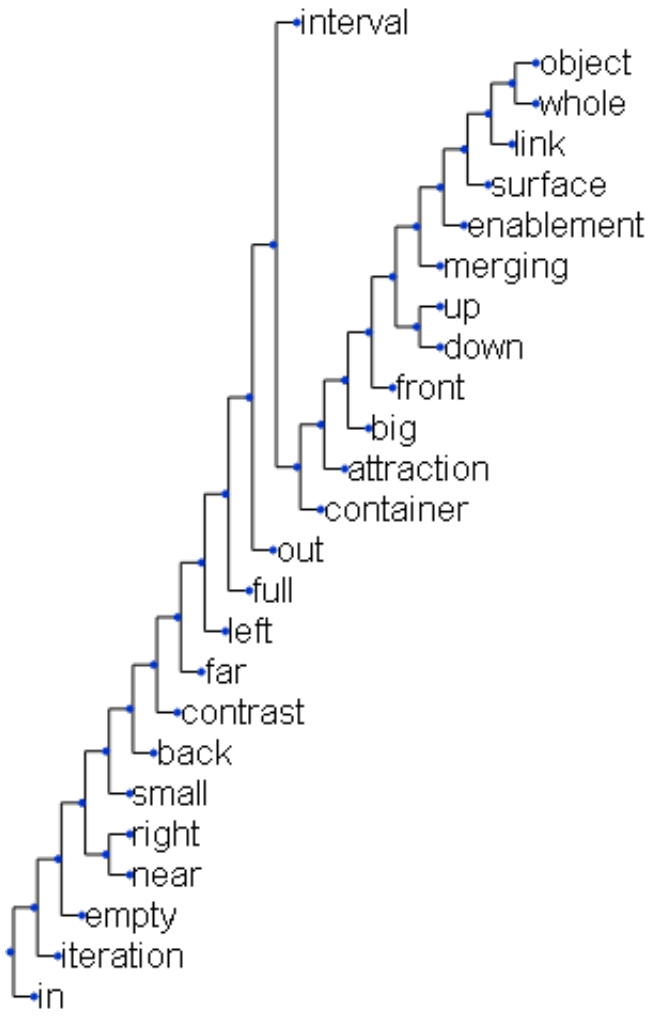

Figure 6.8 – The hierarchical clustering of image schemas in BERT and cosine distance (between two vectors) with the centroid distance (between two image schemas)

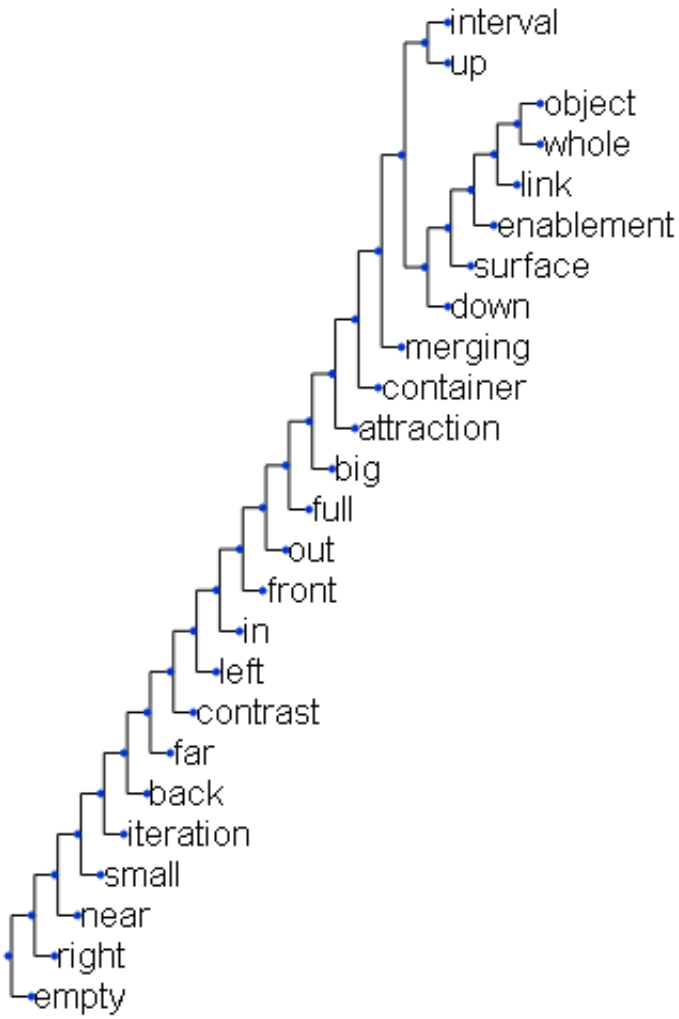

Figure 6.9 – The hierarchical clustering of image schemas in SenseBERT and euclidean distance (between two vectors) with the centroid distance (between two image schemas)

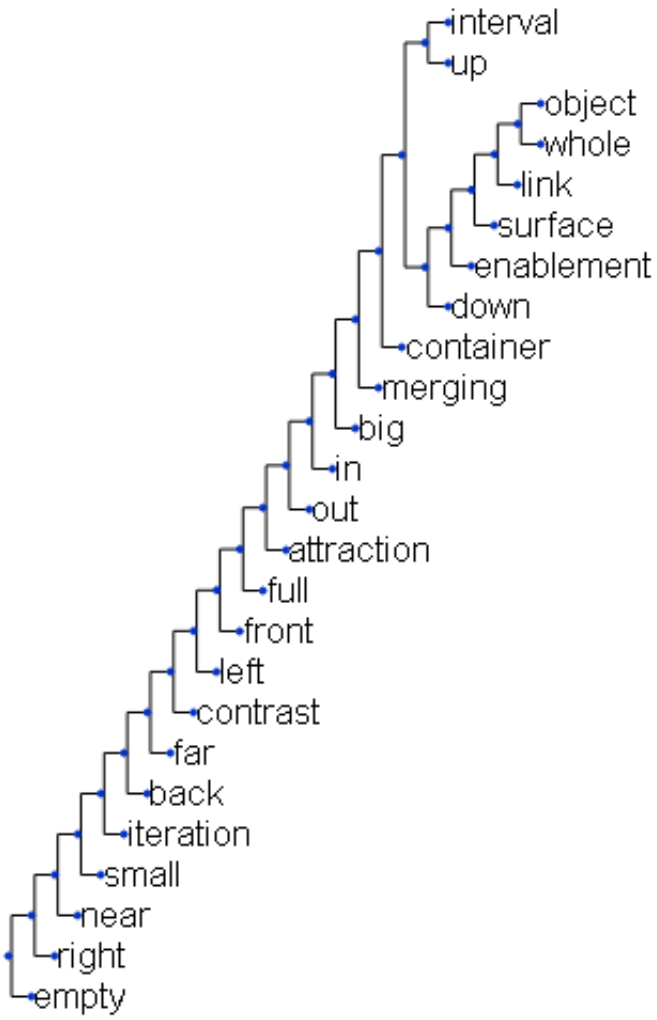

Figure 6.10 – The hierarchical clustering of image schemas in SenseBERT and cosine distance (between two vectors) with the centroid distance (between two image schemas)

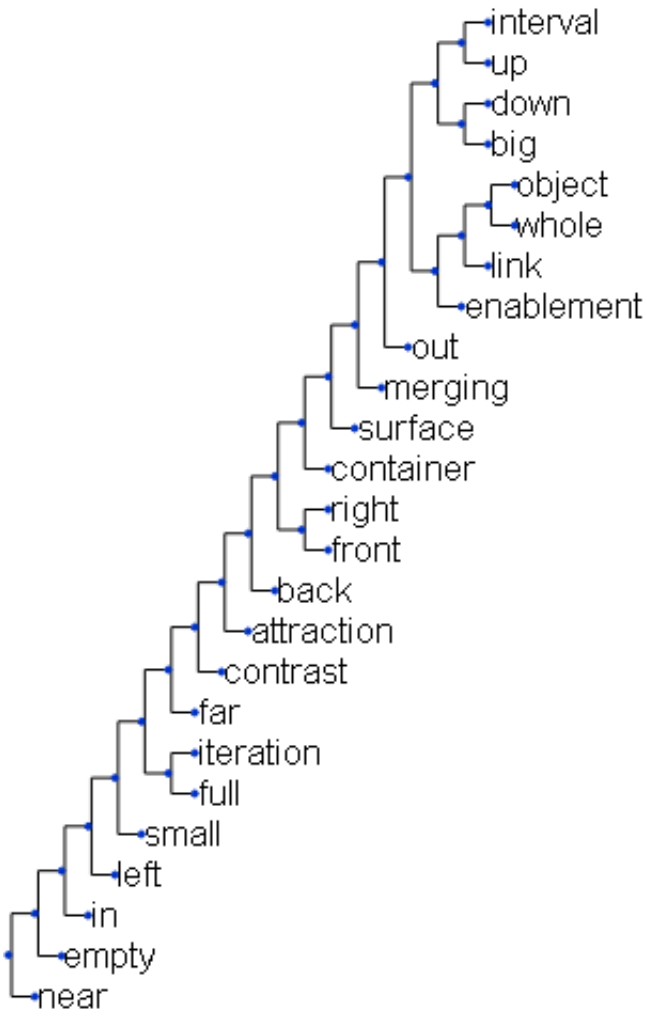

Figure 6.11 – The hierarchical clustering of image schemas in BERT and euclidean distance (between two vectors) with the confusion distance (between two image schemas)

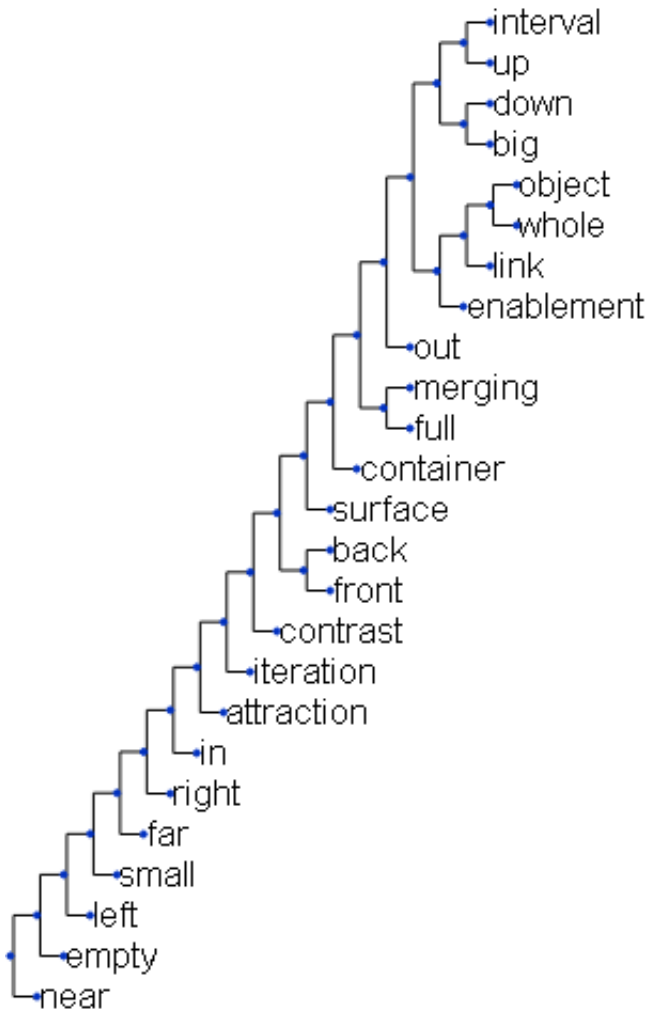

Figure 6.12 – The hierarchical clustering of image schemas in BERT and cosine distance (between two vectors) with the confusion distance (between two image schemas)

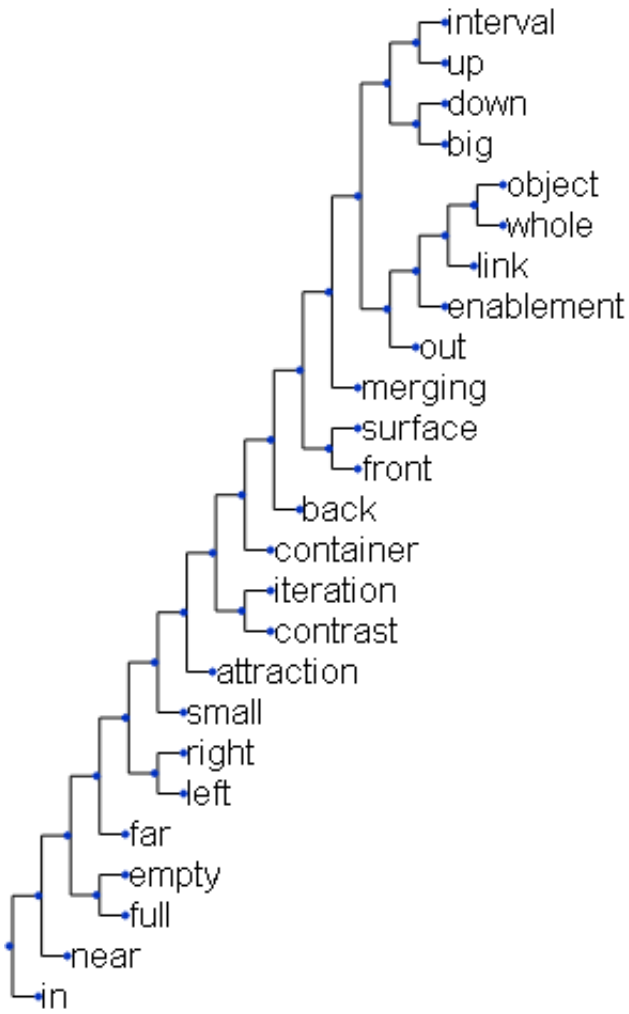

Figure 6.13 – The hierarchical clustering of image schemas in SenseBERT and euclidean distance (between two vectors) with the confusion distance (between two image schemas)

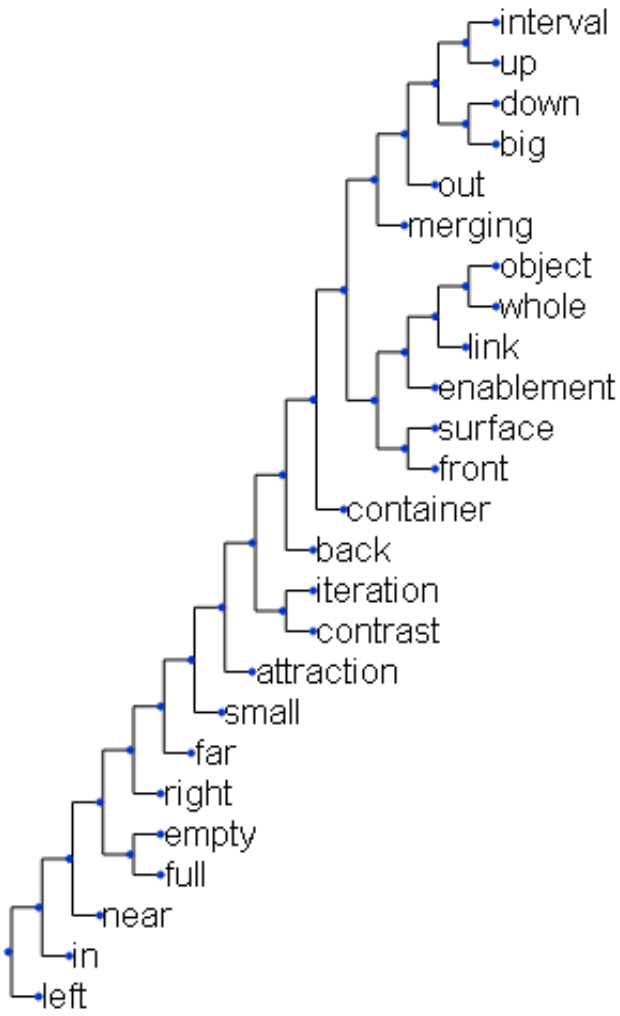

Figure 6.14 – The hierarchical clustering of image schemas in SenseBERT and cosine distance (between two vectors) with the confusion distance (between two image schemas)

which suggest that they are a relative outlier among all the vectors. It should be noted that due to the large number of the vectors, we cannot run the UMAP algorithm with our entire data. Therefore, we sample the data to make the computation tractable.

6.7 Discussion

We can see from Tables 6.7, 6.8, 6.9, and 6.10, SenseBERT embeddings show a better clustering behavior. It holds true whether we measure the clustering by using inertia or by using F_1 score. It also holds true whether we measure the distance by using euclidean distance or by using cosine distance. It also holds true whether the vector values are obtained by averaging the BERT/SenseBERT embedding vectors of all the words in the phrase or by taking the BERT/SenseBERT embedding vector of the word which is tagged with the image schema. This is likely caused by the fact that SenseBERT is explicitly trained by using WordNet, which the image schema computation also uses. Therefore, if we want to represent an image schema as a vector, SenseBERT is more suitable as the base than BERT is. We can use the centroids of the clusters as the embedding vector for each image schema.

We also observe in the same four tables that the inertias and the F_1 scores are higher when we take the vector values from only the word which is tagged with the image schema instead of when we average all the words in the phrase. This is likely caused by the fact that the rests of the phrase are quite similar across various image schemas, which cause the differences between different image schemas (i.e. different clusters) become less pronounced than when we take only the words which are tagged with the image schemas.

We can see the lists of the closest pairs of image schemas in Tables 6.11, 6.12, 6.13, and 6.14. Interestingly, OBJECT-WHOLE is the closest pair in all cases. Besides that, we can see in Table 6.6 that OBJECT and WHOLE together form a large majority of our vectors: more than 70%. It suggests that both image schemas are too general and should be split and refined further.

We can also see in Tables 6.11 and 6.12 that both BERT and SenseBERT give the same top-five closest pairs (albeit in a slightly different order) when we use the centroid distance to measure the distance between two image schemas. However, either OBJECT or WHOLE (or both) appear in all those pairs of image schemas. We also know that the centroids of OBJECT and WHOLE are close to each other. We also see in the dendrograms of the hierarchical clusters when we use the centroid distance to measure the distance between two image schemas (Figures 6.7, 6.8, 6.9, and 6.10) that OBJECT and WHOLE are merged first, then other image schema is merged there one by one. It suggests that both OBJECT and WHOLE are relatively in the center and other image schemas "'radiate" from there.

On the other hand, we see in Tables 6.13 and 6.14 where we use the confusion distance that the top-five closest pairs are more diverse: not all of them include either OBJECT or WHOLE. We find INTERVAL-UP, BIG-UP, and BIG-DOWN. The difference with the aforementioned case when we use the centroid distance might be caused by the spread of the vectors away from their respective centroid. The centroids of INTERVAL and UP are likely to be not so near, but there are many of INTERVAL and UP vectors which are near to the centroid of the other image schema. These results also suggest that, those pairs of image schema are likely often "confused with each other" and probably they should be split and refined further.

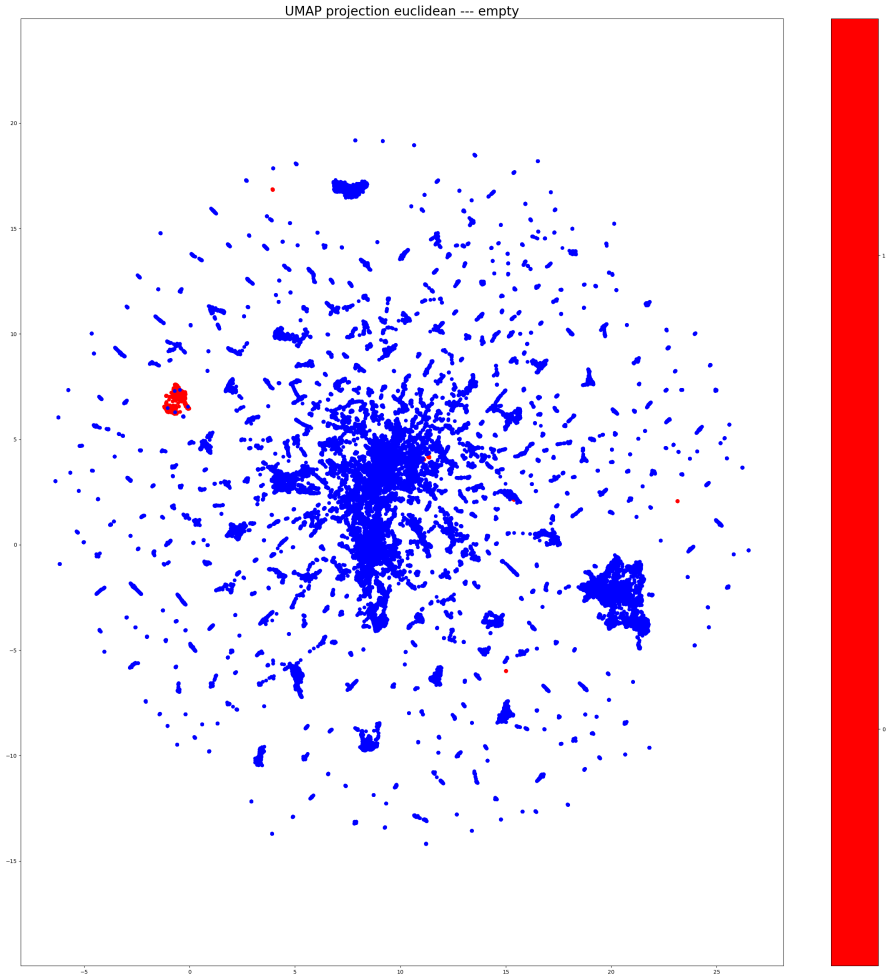

Figure 6.15 – The visualization of the EMPTY image schema (red) and everything else (blue) in BERT and euclidean distance (between two vectors)

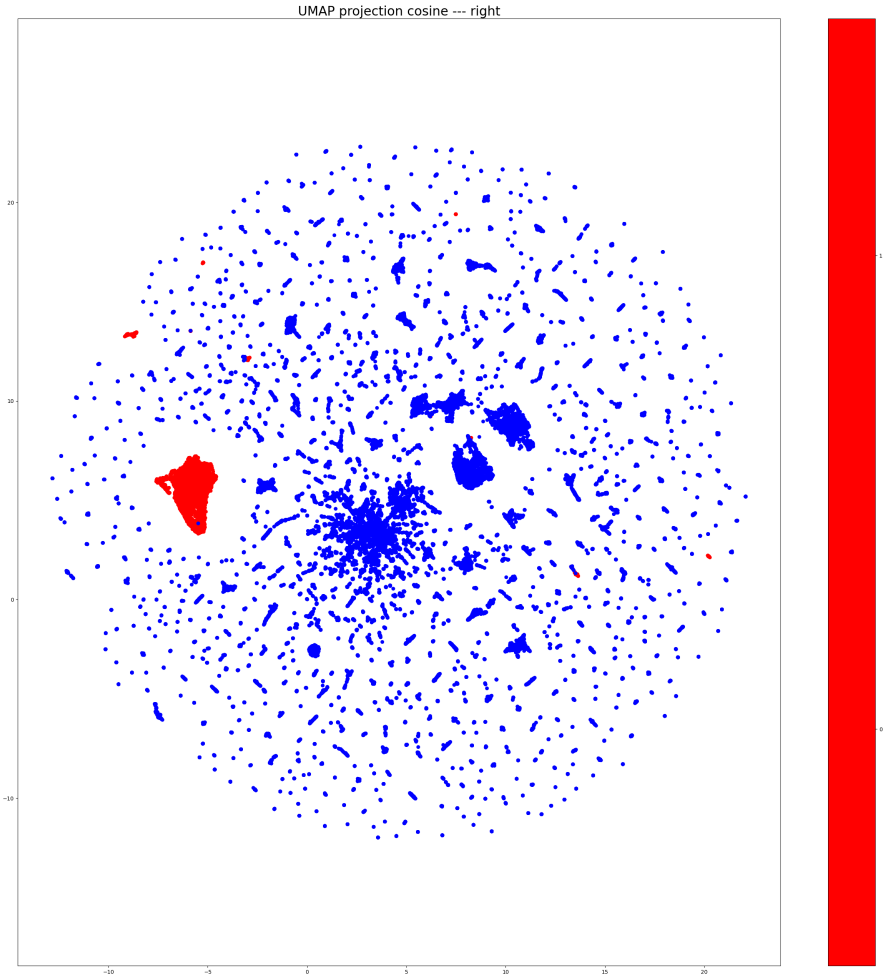

Figure 6.16 – The visualization of the RIGHT image schema (red) and everything else (blue) in SenseBERT and cosine distance (between two vectors)

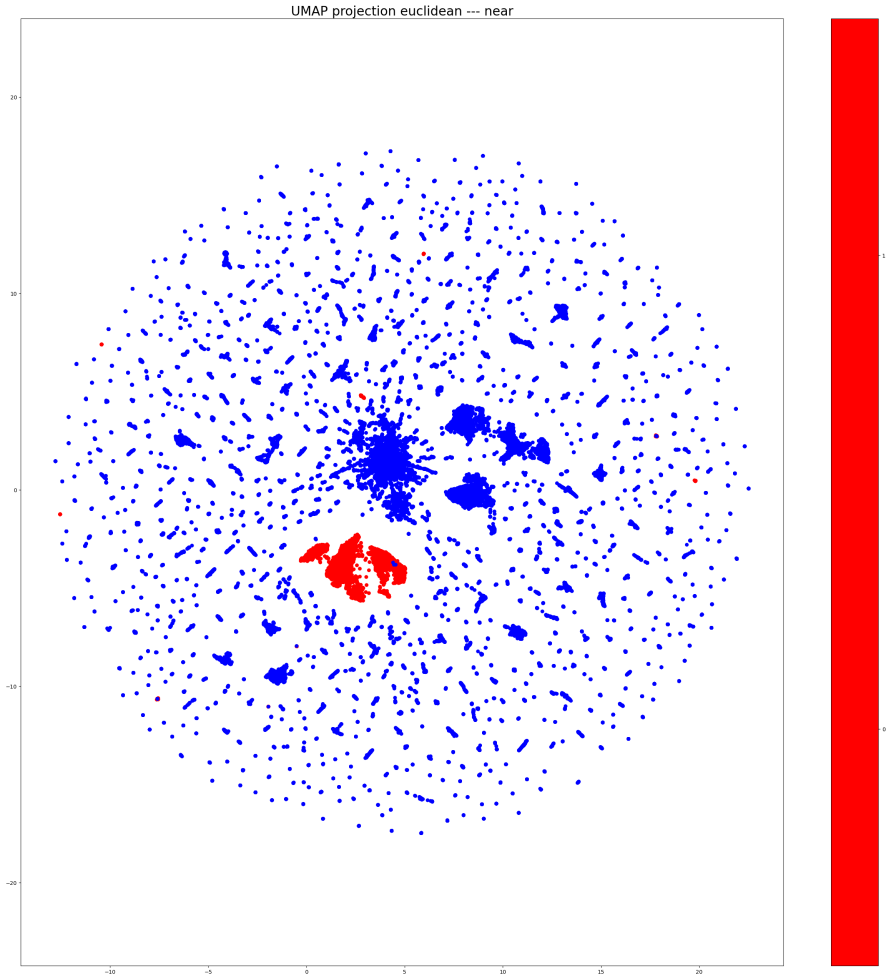

Figure 6.17 – The visualization of the NEAR image schema (red) and everything else (blue) in SenseBERT and euclidean distance (between two vectors)

We also see in the dendrograms of the hierarchical clusters when we use the confusion to measure the distance between two image schemas (Figures 6.11, 6.12, 6.13, and 6.14) that there are many cases of image schemas other than OBJECT and WHOLE merged quite early: INTERVAL-UP (Figures 6.11, 6.12, 6.13, and 6.14), BIG-DOWN (Figures 6.11, 6.12, 6.13, and 6.14), MERGING-FULL (Figure 6.12), and SURFACE-FRONT (Figures 6.13 and 6.14). These results are more diverse than when we use the centroid distance where the hierarchical clusters show that the OBJECT and WHOLE are to be merged together and then other image schemas are to be merged into the OBJECT-WHOLE cluster one by one. The more diverse results when we use the confusion distance is likely because the confusion distance takes into account the vectors which are far from their respective centroid and there are many such vectors.

Finally, we show three UMAP visualizations: the EMPTY image schema when we use BERT and euclidean distance (Figure 6.15), the RIGHT image schema when we use SenseBERT and cosine distance (Figure 6.16) and the NEAR image schema when we use SenseBERT and euclidean distance (Figure 6.17). In all the three cases, those image schemas are among the last to be merged in their respective hierarchical clustering when we use the centroid distance (Figure 6.7 for EMPTY, Figure 6.10 for RIGHT, and Figure 6.9 for NEAR). The fact that they are among the last to be merged suggests that they are a relative outlier, and indeed we see in the three UMAP visualizations that they are away from the dense center.

Here we also give some examples on sentences and their image schemas:

- The men are fighting outside a deli. → "deli" is "WHOLE"
- Two kids in numbered jerseys wash their hands. → "hands" is "OBJECT"
- The woman's hands are empty. → "empty" is "EMPTY"
- A woman is talking on the phone while standing next to a dog. → "next" is "RIGHT"

6.8 Conclusion

Johnson (2013) proposes an abstract notion of image schema as a way to represent semantics. Cienki (2013) proposes the notion that image schema is the way semantics is represented in order to produce metaphoric gestures. Lücking et al. (2016); Cienki (2008) do experiments to investigate the concordance between image schemas and observed gestures. However, although their studies suggest that image schema are indeed relevant for metaphoric gestures, it was not known yet how to extract the image schema from natural communication. Ravenet et al. (2018a,b) propose an algorithm to solve that problem. Ravenet et al's algorithm extract the image schema from a free-form text by using WordNet senses. However, Ravenet et al's algorithm does not give a notion of how similar two different image schemas are: OBJECT, WHOLE, and EMPTY are simply different image schemas. Besides that, recent machine learning techniques which take text as the input (or one of the inputs) require that the text to be converted into vectors. In the wider machine learning world, this is done by using word embedding techniques to convert the text into vectors. These word embedding techniques have a property that two similar words are mapped into two vectors which are close to each other, but different word embedding techniques have different notions on what makes two words to be similar. Incidentally, with word embedding vectors, it is also possible to quantify how similar two words are,

because the distance between the vectors is a proxy of the similarity between the words. Our work has the purpose of representing image schemas as vectors.

We extract the vectors of the individual words in our corpus by using BERT and SenseBERT. We extract the image schemas by running Ravenet et al's algorithm. However, there are far more words than image schemas, thus many words (with different word embedding vectors) are necessarily mapped to the same image schema. However, because they should have similar meanings, then they should be close to each other, which means the embedding vectors which belong to the same image schema should form a cluster. We investigate the clustering behavior when we use BERT and SenseBERT. We also find out which pairs of image schemas are close to each other. Related to this closeness between pairs of image schemas, we show the hierarchical clusters which signifies the relative distance between different image schemas.

Chapter 7

Gesture Shape Representation and Image Schema

In this chapter, we introduce our representation scheme of the gesture shapes. Our objective is to represent gesture shapes such that we can represent wide-enough variety of meanings but is compact enough such that it can be practically encoded. This is similar in principle to the works we discuss in Section 2.5. Specifically, we create the representations based on our observations of the metaphoric gestures used in the Gest-IS corpus (Saint-Amand (2018), *see* Capter 4). After we describe the representation scheme in Section 7.1, we discuss the general statistics of the gestures in Section 7.2. Finally, we relate the gesture shapes and the image schemas (see Section 2.6 and Chapter 6) and we show the statistics in Section 7.3.

7.1 Gesture Shape Representation

The recent neural-network based techniques (Hasegawa et al. (2018); Kucherenko et al. (2019); Ferstl et al. (2019); Kucherenko et al. (2020); Ahuja and Morency (2019)) use 3D coordinates of the joints to represent the gesture. They do not differentiate the hand gestures from the more general body movement. This representation is practical for machine-learning-based techniques because this representation is readily in the form of vector of real numbers, which is what machine learning techniques work with. This representation is also practical in a different way because this data can be extracted efficiently if both Motion Capture instrument and actors are available. However, the motion capture instrument is expensive. Besides that, the process is still laborious: the actors have to be fitted with sensors, the instrument has to be configured, and the actors have to perform various movements. Effectively, it makes obtaining the data an expensive endeavor. Ginosar et al. (2019); Ahuja et al. (2020) circumvent this problem by working directly with 2D data instead of 3D data. They extract the 2D skeletal key point data from videos by using OpenPose (Cao et al. (2017)). OpenPose itself is a software for 2D pose estimation. Due to the wide availability of videos (e.g. from video sharing websites), their data is cheaper to obtain than the traditional method of using Motion Capture instrument. Because their techniques work directly with 2D data, this technique also yields the 2D skeletal key points as the output. This is not a problem because their outputs are 2D videos, and therefore 2D skeletal key points are sufficient. Habibie et al. (2021), on the other hand, automatically

Our Term	Similar ASL Shape	Picture
Circle	O	Figure 7.1
Fist	S	Figure 7.2
Open	5	Figure 7.3
Pinch	Q	Figure 7.4
Point	1	Figure 7.5
Relax	C	Figure 7.6
Two	V	Figure 7.7

Table 7.1 – Our gesture shape representation on the hand shapes, both in our term and the similar shape in the American Sign Language (ASL)

Figure 7.1 – The hand shape “circle” (see Table 7.1)

infer the 3D key points from the 2D key points. Specifically, Habibie et al use Mehta et al. (2020) to infer the body's 3D key points, Zhou et al. (2020) to infer the 3D key points of the hands, and Garrido et al. (2016) to infer the facial 3D key points.

However, as we explain in Section 2.5, Efron (1941); Kipp et al. (2007); Lücking et al. (2016); Calbris (2011) have different representations of gestures. These representations are at a “higher level”. All of them encode the body parts which do the gesture, the trajectory, and the movement direction. We also observe that several ideational gesture generation techniques also use a “higher-level representation” instead of the coordinates of the joints. For example, Bergmann and Kopp (2009) represent the gesture by the “representation technique”, “handedness”, “handshape”, “palm orientation”, “finger orientation”, “movement orientation”, and “movement”. Nihei et al. (2019) classify gesture forms into “square”, “vertical rectangle”, “horizontal rectangle”, “circle”’, “vertical ellipse”, “horizontal ellipse”, and “line shape”. Therefore, in the spirit of generating metaphoric gestures, we also create a “higher-level representation” of gesture shapes. Our representation scheme is inspired by the ones in Kipp et al. (2007) and Lücking et al. (2016). However, our representation scheme is based on our observations in the Gest-IS corpus (Saint-Amand (2018), see Chapter 4).

Our representation consists of the hand shape, the movement type, the movement direction, the movement count (whether it is one way only or with repetition), and the palm orientation. The list of the hand shapes is furnished in Table 7.1. Following Kipp et al. (2007); Lücking et al. (2016) who use the American Sign Language notation, we also furnish the similar shapes in the American Sign Language. The list of movement types is furnished in Table 7.2. The list of movement directions is furnished in Table 7.3. The movement count list is furnished in Table 7.4. Lastly, we furnish the palm orientation list in Table 7.5. It should be noted that palm orientation is different from the movement direction. Palm orientation is the direction where the palm is pointing to whereas the movement direction is the direction where the hand is moving to.

Compared to the representation schemes of Kipp et al. (2007) and Lücking et al. (2016), we recognize less hand shapes. This is deliberate because we create our anno-

Figure 7.2 – The hand shape “fist” (see Table 7.1)

Figure 7.3 – The hand shape “open” (see Table 7.1)

Figure 7.4 – The hand shape “pinch” (see Table 7.1)

Figure 7.5 – The hand shape “point” (see Table 7.1)

Figure 7.6 – The hand shape “relax” (see Table 7.1)

Figure 7.7 – The hand shape “two” (see Table 7.1)

Name	Picture
Linear	Figure 7.8
Circular	Figure 7.9
Waving	Figure 7.10

Table 7.2 – Our gesture shape representation on the movement type

Figure 7.8 – An example of the movement type “linear” (see Table 7.2)

Figure 7.9 – An example of the movement type “circular” (see Table 7.2)

Figure 7.10 – An example of the movement type “waving” (see Table 7.2)

Direction	Picture	Applicable to movement type
Backward	Figure 7.12	Linear
Frontward	Figure 7.12	Linear
Upward	Figure 7.11	Linear
Downward	Figure 7.11	Linear
Inward	Figure 7.11	Linear
Outward	Figure 7.11	Linear
Vertical	Figure 7.15	Circular
Wrist rotation	Figure 7.13	Waving
Horizontal	Figure 7.14	Waving
Normal waving	Figure 7.16	Waving

Table 7.3 – Our gesture shape representation on the movement direction

Name	Applicable to movement type
One Way	Linear and Circular
With Repetition	Linear and Circular

Table 7.4 – Our gesture shape representation on the movement count. “With Repetition” means that the movement is back-and-forth, like a pendulum

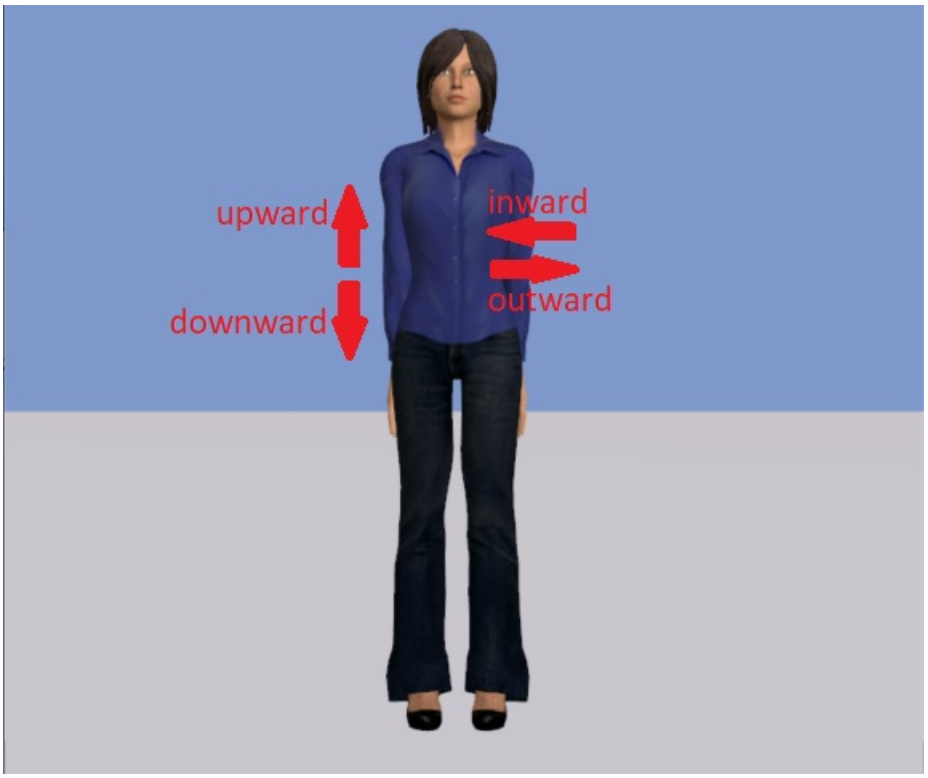

Figure 7.11 – The “inward”, “outward”, “upward”, and “downward” movement directions (see Table 7.3)

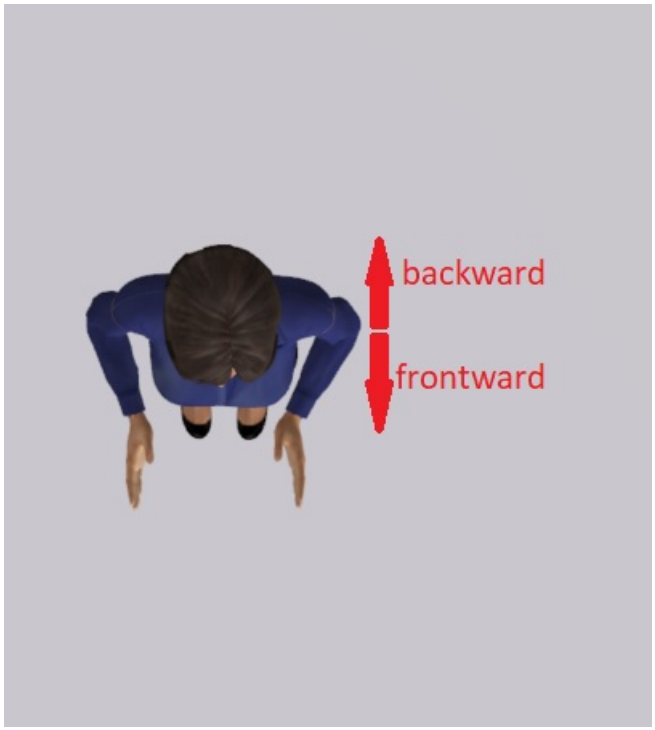

Figure 7.12 – The “frontward” and “backward” movement directions (see Table 7.3)

Figure 7.13 – The “wrist rotation” movement direction (see Table 7.3). In this movement direction, the only movement is the wrist rotation

Figure 7.14 – The “horizontal” movement direction (see Table 7.3). In this movement, the section of the arm between the elbow and the fingertips is moving left-and-right only

Figure 7.15 – The “vertical” movement direction (see Table 7.3). In this movement, the hand is making a half-circle movement

Figure 7.16 – The “normal waving” movement direction (see Table 7.3). The movement is like doing a normal hand-waving movement (like the gesture while we are saying “good bye” to someone). This movement is done by elbow rotations

Orientation	Picture
Backward	Figure 7.17
Frontward	Figure 7.18
Upward	Figure 7.19
Downward	Figure 7.20
Inward	Figure 7.21
Outward	Figure 7.22

Table 7.5 – Our gesture shape representation on the palm orientation

Figure 7.17 – The “backward” palm orientation (see Table 7.5)

Figure 7.18 – The “frontward” palm orientation (see Table 7.5)

Figure 7.19 – The “upward” palm orientation (see Table 7.5)

Figure 7.20 – The “downward” palm orientation (see Table 7.5)

Figure 7.21 – The “inward” palm orientation (see Table 7.5)

Figure 7.22 – The “outward” palm orientation (see Table 7.5)

tation scheme based on what we observe in the Gest-IS corpus, and therefore we only recognize what we observe there. Besides that, we want a compact representation, therefore we want to keep the scheme as “small” as possible. On the classification of the movement type, our scheme, Kipp et al's scheme, and Lücking et al's scheme are different from each other. Our scheme recognizes linear, circular, and waving movement types. Kipp et al's scheme recognizes straight and curved movement types. Meanwhile, Lücking et al's scheme recognizes line, arc, zigzag, and pointing movement types.

7.2 Overall Statistics

We extract the metaphoric gesture strokes from the Gest-IS corpus (Saint-Amand (2018), see Chapter 4). In this section, we give the statistics pertaining to the shape of those gestures.

In Tables 7.6 and 7.7 where we furnish the statistics about the starting and the ending hand shapes, we observe in the two tables that the “Open” hand shape forms the majority, despite the fact that these are metaphoric gestures. Thus, even metaphoric gestures, which are supposed to convey abstract ideas, are mostly performed with a simple hand shape. Interestingly, the second most common hand shape is “Relax”, which is also a simple hand shape.

In Tables 7.8 and 7.9 where we furnish the starting and ending palm orientations, we observe that the two most common palm orientations are “Downward” and “Inward”, but they are not extremely numerous compared to the other palm orientations. It might be

Starting Hand Shape (see Table 7.1)	Count	Proportion
Open	99	68.75%
Relax	20.5	14.236%
Fist	12.5	8.681%
Pinch	6	4.167%
Point	3	2.083%
Circle	2	1.389%
Two	1	0.694%

Table 7.6 – The starting hand shape counts. The non-integer numbers mean that the shapes are different for the left hand and the right hand, where each of them counts as 0.5

Ending Hand Shape (see Table 7.1)	Count	Proportion
Open	110.5	76.736%
Relax	18.5	12.847%
Pinch	6	4.167%
Fist	3	2.083%
Point	3	2.083%
Circle	2	1.389%
Two	1	0.694%

Table 7.7 – The ending hand shape counts. The non-integer numbers mean that the shapes are different for the left hand and the right hand, where each of them counts as 0.5

caused by the fact that the speakers in our corpus are sitting. Having the palms facing inward or downwards and the hands are resting on the thighs is a comfortable resting position while sitting. Therefore, these orientations are probably "carried" when the gestures are performed.

In Table 7.10 where we furnish the movement types, we observe that the "linear" gestures are by far the most common. In Table 7.13, we find that the vast majority of those "linear" gestures are one way. This is the simplest movement type. Interestingly, we also saw earlier that the most common hand shapes are the simplest ones too. Thus, most of the gestures are probably relatively simple.

In Table 7.11 where we furnish the movement direction of the "linear" gestures, the most common one is "downward". However, the "downward" movement does not greatly outnumbers the others. This might be related to the fact that the most common palm orientation is also "downward" (see Tables 7.8 and 7.9).

7.3 Statistics Pertaining To Gesture Shape and Image Schema

We extract the samples from our corpus in a way similar to when we predict gesture timing (see Section 5.1): we split the samples according to the Inter-Pausal Unit. However, we only take those with a metaphoric gesture. We run them through the algorithm to extract the image schemas (see Sections 6.3.1 and 6.4). We only take the samples which have an

Starting Palm Orientation (see Table 7.5)	Count	Proportion
Downward	54.5	37.847%
Inward	36	25%
Backward	30.5	21.181%
Frontward	17	11.806%
Upward	6	4.167%

Table 7.8 – The starting palm orientation counts. The non-integer numbers mean that the orientations are different for the left hand and the right hand, where each of them counts as 0.5

Ending Palm Orientation (see Table 7.5)	Count	Proportion
Downward	45	31.25 %
Inward	35	24.306%
Upward	31	21.528%
Frontward	18	12.5%
Backward	15	10.417%

Table 7.9 – The ending palm orientation counts. The non-integer numbers mean that the orientations are different for the left hand and the right hand, where each of them counts as 0.5

Movement Type (see Table 7.2)	Count	Proportion
Linear	133	92.361%
Waving	7	4.861%
Circular	4	2.778%

Table 7.10 – The movement type counts

Direction (see Table 7.3)	Count	Proportion
Downward	46	34.586%
Outward	29	21.805%
Frontward	20	15.038%
Upward	18	13.534%
Inward	16	12.03%
Backward	4	3.008%

Table 7.11 – The movement direction counts of the “linear” movement type (see Table 7.2)

Direction (see Table 7.3)	Count	Proportion
Normal Waving	4	57.143%
Horizontal	2	28.571%
Wrist Rotation	1	14.286%

Table 7.12 – The movement direction counts of the “waving” movement type (see Table 7.2)

Type (see Table 7.4)	Count	Proportion
One Way	126	94.737%
Many	7	5.263%

Table 7.13 – The statistics of the movement count (i.e. one way or many) of the ”linear” movement type (see Table 7.2)

image schema. The counts of the samples for each image schema are furnished in Table 7.14.

It can be seen in Table 7.14 that “Object” and “Whole” image schemas form the majority. Therefore, we will investigate these two image schemas. We furnish the counts of the ending hand shape and the ending palm orientation. For the movement, because we observe in the general statistics (see Tables 7.10 and 7.13) that one-way linear movements form the majority, we will investigate the difference of the movement directions of the one-way linear movements.

In Tables 7.15 and 7.16 where we show the ending hand shape counts of the “Object” and “Whole” image schemas, it can be seen that “Open” is the most common hand shape. This is similar to what we observe in the general case (see Table 7.7).

For the ending palm orientations of the “Object” and “Whole” image schemas, as we show in Tables7.17 and 7.18, we find that in both image schemas “upward”, “downward”, and “inward”. This is similar to what we find in the general case (see Table 7.9).

And finally, for the movement directions of the linear one-way movements, as we furnish in Tables 7.19 and 7.20, in both “Object” and “Whole” image schemas, both downward and outward movement directions are relatively numerous, but not by a large margin compared to the others. This is similar to what we see in the general case (see Table 7.11).

Image Schema	Count	Proportion
Object	42	45.652%
Whole	31	33.696%
Big	7	7.609%
Merging	3	3.261%
Link	3	3.261%
Attraction	3	3.261%
Up	2	2.174%
Interval	1	1.087%

Table 7.14 – The counts of the samples with an image schema

Ending Hand Shape (see Table 7.1)	Count	Proportion
Open	33	78.571%
Relax	6	14.286%
Circle	2	4.762%
Pinch	1	2.381%

Table 7.15 – The ending hand shape counts of the "Object" image schema

Ending Hand Shape (see Table 7.1)	Count	Proportion
Open	24.5	79.032%
Relax	3.5	11.290%
Pinch	2	6.452%
Point	1	3.226%

Table 7.16 – The ending hand shape counts of the "Whole" image schema. The non-integer numbers mean that the hand shape are different for the left hand and the right hand, where each of them counts as 0.5

Ending Palm Orientation (see Table 7.5)	Count	Proportion
Inward	15	35.714%
Downward	10	23.81%
Upward	8	19.048%
Frontward	6	14.286%
Backward	3	7.143%

Table 7.17 – The ending palm orientation counts of the "Object" image schema

Ending Palm Orientation (see Table 7.5)	Count	Proportion
Downward	11	35.484%
Upward	6	19.355%
Inward	6	19.355%
Backward	4	12.903%
Frontward	4	12.903%

Table 7.18 – The ending palm orientation counts of the "Whole" image schema

Direction (see Table 7.3)	Count	Proportion
Upward	8	23.529%
Downward	7	20.588%
Outward	7	20.588%
Frontward	6	17.647%
Inward	6	17.647%

Table 7.19 – The movement direction counts of the "Object" image schema for the one-way linear movements

Direction (see Table 7.3)	Count	Proportion
Downward	8	29.63%
Outward	7	25.926%
Frontward	5	18.519%
Inward	3	11.111%
Upward	3	11.111%
Backward	1	3.704%

Table 7.20 – The movement direction counts of the "Whole" image schema for the one-way linear movements

We observe that these two image schemas have gesture shapes which are not only similar to the overall statistics, but they are also similar to each other (i.e. "Object" and "Whole").

7.4 Conclusion

In this chapter, we define our gesture shape representation scheme. We are inspired by the gesture representation scheme in Kipp et al. (2007) and Lücking et al. (2016), but we create our scheme according to our observations in the Gest-IS corpus (Saint-Amand (2018)). Our representation consists of the hand shape, the movement type, the movement direction, the movement count (whether it is one way only or with repetition), and the palm orientation.

We discuss in Section 7.2 the general statistics of the gesture shapes. We find that most gestures are simple gestures, both in terms of the hand shape and in movement. It might be related to the fact that our corpus contains natural conversations instead of a staged performance (see Chapter 4). Interestingly, despite the fact that our representation scheme, Kipp et al's representation scheme, and Lücking et al's representation scheme have different classifications of the movement type, ultimately what we observe is that most of the movements are linear/straight, which is recognized by all of the three schemes.

Finally, in Section 7.3 where we relate the gesture shapes and image schemas, we find that "Object" and "Whole" account for most of the image schemas. The gesture shapes of these two image schemas are similar to the overall statistics. Besides that, the gesture shapes of these two image schemas are also similar to each other. It is probably related to our finding in Section 6.7 that the image schemas "Object" and "Whole" are indeed similar to each other.

Chapter 8

Conclusion

Our work contributes toward the general problem of communicative gesture generation for virtual agent.

In Chapters 4 and 7, we explain our corpus and the additional data extraction/annotation we perform. We extract the acoustic features, eye brow movements, and metaphoric gesture shapes. In order to annotate the metaphoric gesture shapes, we define our gesture shape annotation scheme.

In Chapter 5, we explain our model to predict gesture timing. This work is a step towards generating gestures with the desired property that the gestures should match/be coherent with the speech (see Section 1.3.1), especially toward generating gestures whose timing matches the speech's rhythm. Two important developments in this domain which are reflected in our approach are the use of machine learning and the use of acoustic features (see Sections 1.4.1 and 1.4.2).

In Chapter 6, we explain our proposal to improve the existing image schema computation algorithm and we also explain our proposed method to represent image schemas as embedding vectors. This work is a step toward generating gestures with the desired property that the gestures should match/be coherent with the speech (see Section 1.3.1), especially toward generating gestures whose shape matches the speech's semantics. Three important developments in this domain which are reflected in our approach are the use of machine learning, word embedding, and image schema (see Sections 1.4.1, 1.4.6, and 1.4.7). To enable the generation of gestures whose shape matches the speech's semantics, we have to encode the semantics. Image schema is one way to encode such semantics. An interesting property of image schema is that its relevance is supported by prior works which find the relationship between image schema and metaphoric gestures (see Section 2.6). However, neural network takes only vectors as its input. In order to solve this problem, word embedding techniques have been developed and have been used to transform texts into vectors. Therefore, we develop a method to transform image schemas into embedding vectors.

In this closing chapter, we explain our contributions (Section 8.1) and their limitations (Section 8.2). We also explain the potential future works and how they fit into the broader context of the gesture generation problem (Section 8.3).

8.1 Contribution Summary

In this section, we summarize our contributions. Our contributions are the corpus annotation and analysis in Section 8.1.1, prediction of gesture timing in Section 8.1.2, improvement of Ravenet et al. (2018a,b)'s algorithm in Section 8.1.3, and representation of image schema as vectors in Section 8.1.4.

8.1.1 Corpus Annotation and Analysis

The Gest-IS corpus (Saint-Amand (2018)) we use comes with the transcripts and the annotations of communicative or non-communicative gestures, the gesture types, and the gesture phases. It is, however, possible to extract additional data from the corpus. We extract the eyebrow movements, the acoustic features, and the metaphoric gesture shapes.

We use OpenFace (Baltrusaitis et al. (2018)) to extract the eyebrow movements. We extract 3 Action Units (AUs) which represent the eyebrow movements: AU1 (inner brow raiser), AU2 (outer brow raiser), and AU4 (brow lowerer). With these three AUs, we can detect both rising and lowering eyebrow movements. This contribution can be found in Section 4.5.

To extract the acoustic features, we use OpenSmile (Eyben et al. (2010)). We extract the audio features with 100 milliseconds time-step. We extract the fundamental frequency (F_0), the F_0 direction score, the intensity, and the Mel-frequency cepstral coefficients (MFCC). We use these features in our work on the prediction of gesture timing (see Chapter 5). This contribution can be found in Section 4.7.

To extract the metaphoric gesture shapes, we have to define the representation/annotation scheme beforehand. Our scheme is inspired by the prior works on gesture encoding (see Section 2.5), but we develop our scheme based on our observations in the corpus. Our representation scheme is available in Section 7.1. After that, we analyze the statistics of the different dimensions of the gesture shapes, including the breakdown of such features for different image schemas. The analyses are available in Sections 7.2 and 7.3.

8.1.2 Prediction of Gesture Timing

We develop a neural network model by using recurrent neural network with attention mechanism to predict gesture timing according to the acoustic input. Other than developing the model, we also develop an objective evaluation measure which tolerates shift and dilation, we experiment with including eye brow movements in our data, and we investigate whether the model is generalizable to the conversation partner.

Our model's approach of expressing the problem as a time series prediction problem where the input is expressed as a sequence of acoustic features is similar to the approaches used in the prior works which use acoustic features for gesture generation (see Sections 3.3 and 3.5). We use three prosody features, namely fundamental frequency (F_0), F_0 direction score, and intensity. We also do a separate experiment with Mel-frequency cepstral coefficients (MFCC). The use of these features is also similar to the prior works in which use acoustic features for gesture generations (see Section 3.3 and 3.5). This contribution is available in Section 5.2.2.

We develop an objective evaluation measure which tolerates shift and dilation up to a certain extent. The underlying spirit is similar to the other works which allow many-to-many relationships (e.g. Ginosar et al. (2019); Ferstl et al. (2019); Wu et al. (2021))

which recognizes some diversity on the results (see Section 1.3.3): for each input, there can be multiple correct output. However, our approach is more narrow: we only tolerate differences in the form of shifts and dilations. This contribution is available in Section 5.3.

We do an experiment where we count the eyebrow movements as beat gestures. Our corpus's gesture annotation only includes the hand gestures, which leads us into the question of whether including eyebrow movements as beat gestures will yield better results. Indeed, our validation process becomes more reliable for the beat gestures when we include the eyebrow movements. Gestures and facial movements are traditionally considered as separate problems in virtual agent. However, our finding suggest that these two problems should probably be treated as different facets of the same problem. This contribution is available in Sections 5.4 and 5.5, Experiment 4.

We also investigate whether our model is generalizable to the conversation partner. Our corpus consists of conversations between the same pair of persons. Therefore, we do an experiment to investigate whether a model trained on one speaker performs well when tested on the data of the other speaker, and we find that the model indeed performs well. This contribution is available in Section 5.4 and 5.5, Experiment 7.

8.1.3 Improvement of Ravenet et al. (2018a,b)'s Algorithm

We make two improvements on the Ravenet et al's algorithm (see Section 6.4).

The first improvement is replacing the Lesk algorithm which Ravenet et al use for word sense disambiguation with the first WordNet sense. As a method, choosing the first WordNet sense is very simple, yet Raganato et al. (2017) show that this method performs remarkably well in word sense disambiguation tasks. It also consistently performs better than Lesk algorithm in those tasks. This improvement is available in Section 6.4.1.

The second improvement we make is making use of more types of edge for the traversal of WordNet graph. Ravenet et al's algorithm uses only the hypernym edges for the WordNet graph traversal, but only noun and verb senses are connected through hypernym edges. Thus, we use synonym edges for the traversal of adjective senses. Meanwhile, for adverb, we get the corresponding adjective first, and then we treat this sense as if it were an adjective sense. For verb, we also use the troponym edges. Troponym itself means the manner of doing an action. This contribution is available in Section 6.4.2.

8.1.4 Representation of Image Schemas As Vectors

We develop a method to represent image schemas as vectors. With the image schemas representable as vectors, the notion of distance between different image schemas, which is a proxy of the differences between different image schemas, becomes sensible. Therefore, we also calculate the distances between different image schemas. Both of these two contributions are available in Section 6.5.

To represent the image schemas as vectors, first we extract the image schemas (including which word the image schema is attached to) by using the improved version of Ravenet et al's algorithm (see Section 6.4). Then, we get the word embedding by using a word embedding model. We experiment with both BERT (Devlin et al. (2018)) and SenseBERT (Levine et al. (2020)) word embedding models. However, because the words which share the same image schema should express similar meanings, then they should also be close to each other in the word embedding vector space. Therefore, in order to observe whether there is a concordance between the image schemas and the word embedding vectors, we observe whether the vectors which belong to the same image schema

show a clustering behavior. We measure this clustering behavior. And then, we define the centroids of those clusters as the embedding vectors of the image schema.

With the image schemas being representable as vectors, we are also able to calculate the distances between different image schemas, which is a proxy of the differences between different image schemas. We experiment with both the distances between the cluster centroids and the confusion distances (i.e. proportionally, how many data points are nearer to the other cluster's centroid). With their distances are known, we show which image schemas are close/similar to each other with the hierarchical cluster of the image schemas. Interestingly, we find that "OBJECT" and "WHOLE" image schemas are always close to each other, even though we try different distance metrics.

8.2 Limitations of Our Work

In this section, we explain the limitations of our work. Mainly, we discuss the limitations in the two contributions: prediction of gesture timing (see Section 8.1.2) and representation of image schemas as vectors (see Section 8.1.4).

8.2.1 Prediction of Gesture Timing

The first limitation we have is the small size and the small variation in our data. Our data has only conversations between the same dyad for the total duration of around 50 minutes. Although we find that the model trained only on the data of one speaker can predict the timing of the other speaker, we do not know if the model can work on strangers who are not a party of this conversation. That is, we do not know if our model is generalizable beyond the speakers of this conversation.

The second limitation is our strict distinction between ideational gestures and beat gestures. However, the distinction between them is more like a degree of difference than a binary distinction (McNeill (1992)). That is, it is possible that a gesture is both ideational and beat at the same time. The strict distinction we use is a simplification.

The third limitation is that we do not have the information about the "frequency" of the beat gestures. If in our data/prediction we find that $t = 10$ to $t = 30$ is marked with beat, we do not know if it means one slow beat movement, several fast beat movements, or many very-fast beat movements. This is because we treat beat as if it were without phase.

The fourth limitation is that we treat hand gestures and facial movements separately. As can be seen in Section 5.4 Experiment 4, our validation process becomes more reliable for the beat gestures when we include the eyebrow movements. Gestures and facial movements are traditionally considered as separate problems in virtual agent research. However, our finding suggest that these two problems should probably be treated as different facets of the same problem. Interestingly, a recent work from Habibie et al. (2021) generate both gestures and facial movements together. This new paradigm introduced by Habibie et al might be more suitable for this problem.

8.2.2 Representation of Image Schemas As Vectors

The first two limitations we would like to mention are the sources of our important information: the source of the word embedding and the source of the image schema annotation. We try both BERT and SenseBERT as the word embedding techniques and we find

that that SenseBERT yields a better clustering behavior. However, representation learning is an active research area, so there can be new embedding techniques which show a better concordance with image schemas. On the source of image schema annotation, it comes from the improved version of Ravenet et al's algorithm. As far as we are aware of, it is the only known technique to extract image schemas from free-form texts. However, it is possible that someone else comes up with a different annotation algorithm which might give different results.

The third limitation we would like to highlight is that the similarity between the image schemas are evaluated only through objective measurements. We do not do any subjective study to measure the similarity between different image schemas.

The fourth limitation is the lack of portability to another language. Both BERT and SenseBERT are only for English. Similarly, WordNet is only for English as well. There might be word embedding models similar to BERT and SenseBERT for other languages, there might also be a lexical database like WordNet for other languages, but we do not know if the results would be the same with what we obtain.

The fifth limitation is that the method does not tell how to refine an image schema. Although our method shows that image schemas "OBJECT" and "WHOLE" are similar to each other, our method does not tell how to refine them.

8.3 Future Work

One grand objective we have not accomplished is the computation of ideational gesture shape. We have proposed a method to solve the one of its sub-problems: how to represent the semantics in a vector form in such a way that the two similar semantics are represented by two nearby vectors. However, there are at least two other sub-problems which are not yet addressed.

The first unaddressed sub-problem is how to represent the gesture shapes in such a way that two gestures which convey similar semantics are represented by two nearby vectors. For example, it is possible to do a gesture of pointing to the right by using the thumb of the right hand, the index finger of the left hand, or even by all fingers of the right hand. As a human, we might be able to intuitively recognize all of them as pointing to the right. However, it is a challenge for computers to recognize that they are similar gestures. The existing works which use machine learning represent the gestures simply by the coordinates of the body joints (see Section 1.4.3). This representation is convenient for machine learning because this representation is readily in the form of vector of real numbers. Besides that, these joint coordinates can be extracted by using a Motion Capture tool if both the tool and the actors are available. However, it is not clear how we can go from this representation to be able to say that two gestures are similar. There are also prior works which define their own representation of gesture shapes (see Section 2.5). These representations work on a "higher level", such as the movement trajectory or the hand shape. But it is still not clear how to create a notion that two gestures are similar. If gesture X has the index finger of the right hand points to the right, gesture Y has the index finger of the left hand points to the right, and gesture Z has the index finger of the right hand points to the front, human might be able to intuitively notice that both gestures X and Y are pointing to the right while gesture Z is pointing to the front. Therefore, gesture X is more similar to gesture Y than to gesture Z. But it is not clear how we can go from the representation into being able to make a notion of the similarity. Word embedding was born from a similar problem in natural language

processing. Eventually, one particularly successful approach is by creating a notion that two words are similar if they are surrounded by similar words. Word2Vec, GloVe, and BERT share such philosophy (see Section 6.3.2). However, it is not clear if that approach can be used for gesture. Besides that, while there are many large and publicly-available corpora for natural language processing, it is not the case for gesture.

The second unaddressed sub-problem follows the first one. Once we can represent both the semantics and the gesture shape as vectors, we still have to construct a model/algorithm to compute the gesture shape from the semantics. On this, it is worth remembering what properties are desirable from generated gestures (see Section 1.3). The first desired property is that the gestures should match the speech (see Section 1.3.1). This sub-problem is indeed about the match with the speech. The second desired property is that the movements should be smooth (see Section 1.3.2). In this context, this desired property means the generated ideational gestures should have smooth transitions with beat gestures. The transition to the phase when gesture is not performed should also be smooth. All these mean that care should be taken to ensure that the entire movement is smooth. The third desired property is that the gestures should be diverse (see Section 1.3.3). For example, there are many ways to point to the right. In this context, it means that care should be taken such that the model/algorithm is stochastic even after the training.

Bibliography

Agirre, E., López de Lacalle, O., and Soroa, A. (2014). Random walks for knowledge-based word sense disambiguation. *Computational Linguistics* 40, 57–84

Agirre, E. and Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. 33–41

Ahuja, C., Lee, D. W., Nakano, Y. I., and Morency, L.-P. (2020). Style transfer for co-speech gesture animation: A multi-speaker conditional-mixture approach. In *European Conference on Computer Vision* (Springer), 248–265

Ahuja, C. and Morency, L.-P. (2019). Language2pose: Natural language grounded pose forecasting. In *2019 International Conference on 3D Vision (3DV)* (IEEE), 719–728

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., and Bengio, Y. (2016). End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE), 4945–4949

Baltrušaitis, T., Mahmoud, M., and Robinson, P. (2015). Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)* (IEEE), vol. 6, 1–6

Baltrusaitis, T., Zadeh, A., Lim, Y. C., and Morency, L.-P. (2018). Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)* (IEEE), 59–66

Bartlett, M. S., Littlewort, G. C., Sejnowski, T., and Movellan, J. (2002). A prototype for automatic recognition of spontaneous facial actions. *Advances in neural information processing systems* 15, 1295–1302

Bellman, R. and Kalaba, R. (1959). On adaptive control processes. *IRE Transactions on Automatic Control* 4, 1–9

Bergmann, K. and Kopp, S. (2009). Gnetic–using bayesian decision networks for iconic gesture generation. In *International Workshop on Intelligent Virtual Agents* (Springer), 76–89

Biancardi, B., Cafaro, A., and Pelachaud, C. (2017). Analyzing first impressions of warmth and competence from observable nonverbal cues in expert-novice interactions. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. 341–349

Bolinger, D. (1989). *Intonation and its Uses* (Stanford University Press)

Bowman, S., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 632–642

Bozkurt, E., Yemez, Y., and Erzin, E. (2016). Multimodal analysis of speech and arm motion for prosody-driven synthesis of beat gestures. *Speech Communication* 85, 29–42

Bressem, J. and Ladewig, S. H. (2011). Rethinking gesture phases: Articulatory features of gestural movement?

Calbris, G. (2011). *Elements of meaning in gesture*, vol. 5 (John Benjamins Publishing)

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2019). Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence* 43, 172–186

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7291–7299

Cassell, J., Vilhjálmsson, H., and Bickmore, T. (2001). BEAT: the Behavior Expression Animation Toolkit. In *Computer Graphics Proceedings, Annual Conference Series* (ACM SIGGRAPH)

Cassell, J., Vilhjálmsson, H. H., and Bickmore, T. (2004). Beat: the behavior expression animation toolkit. In *Life-Like Characters* (Springer). 163–185

Chiu, C.-C. and Marsella, S. (2014). Gesture generation with low-dimensional embeddings. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. 781–788

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in neural information processing systems*. 577–585

Cienki, A. (2008). Image schemas and gesture. In *From perception to meaning* (De Gruyter Mouton). 421–442

Cienki, A. (2013). Image schemas and mimetic schemas in cognitive linguistics and gesture studies. *Review of Cognitive Linguistics. Published Under the Auspices of the Spanish Cognitive Linguistics Association* 11, 417–432

Cravotta, A., Busà, M. G., and Prieto, P. (2019). Effects of encouraging the use of gestures on speech. *Journal of Speech, Language, and Hearing Research* 62, 3204–3219

Dermouche, S. and Pelachaud, C. (2016). Sequence-based multimodal behavior modeling for social agents. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction* (ACM), 29–36

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*

Edmonds, P. and Cotton, S. (2001). Senseval-2: overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*. 1–5

Efron, D. (1941). Gesture and environment.

Ekman, P. (1979). About brows: Emotional and conversational signals. In *Human ethology: Claims and limits of a new discipline: contributions to the Colloquium*, eds. M. von Cranach, K. Foppa, W. Lepenies, and D. Ploog (Cambridge, England; New-York: Cambridge University Press). 169–248

Ekman, P. (1982). Methods for measuring facial action. *Handbook of methods in nonverbal behavior research* , 45–90

Eyben, F., Wöllmer, M., and Schuller, B. (2010). Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia* (ACM), 1459–1462

Ferstl, Y. and McDonnell, R. (2018). Investigating the use of recurrent motion modelling for speech gesture generation. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*. 93–98

Ferstl, Y., Neff, M., and McDonnell, R. (2019). Multi-objective adversarial gesture generation. In *Motion, Interaction and Games*. 1–10

Feyereisen, P. (1987). Gestures and speech, interactions and separations: A reply to mcneill (1985).

Flecha-García, M. L. (2007). Non-verbal communication in dialogue: Alignment between eyebrow raises and pitch accents in english. In *Proceedings of the Annual Meeting of the Cognitive Science Society*. vol. 29

Friesen, E. and Ekman, P. (1978). Facial action coding system: a technique for the measurement of facial movement. *Palo Alto* 3

Garrido, P., Zollhöfer, M., Casas, D., Valgaerts, L., Varanasi, K., Pérez, P., et al. (2016). Reconstruction of personalized 3d face rigs from monocular video. *ACM Transactions on Graphics (TOG)* 35, 1–15

Garrod, S. and Pickering, M. J. (2009). Joint action, interactive alignment, and dialog. *Topics in Cognitive Science* 1, 292–304

Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2414–2423

Ginosar, S., Bar, A., Kohavi, G., Chan, C., Owens, A., and Malik, J. (2019). Learning individual styles of conversational gesture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3497–3506

Goodwyn, S. W., Acredolo, L. P., and Brown, C. A. (2000). Impact of symbolic gesturing on early language development. *Journal of Nonverbal behavior* 24, 81–103

Graziano, M. and Gullberg, M. (2018). When speech stops, gesture stops: Evidence from developmental and crosslinguistic comparisons. *Frontiers in psychology* 9, 879

Habibie, I., Xu, W., Mehta, D., Liu, L., Seidel, H.-P., Pons-Moll, G., et al. (2021). Learning speech-driven 3d conversational gestures from video. *arXiv preprint arXiv:2102.06837*

Halliday, M. A. K. (1973). Explorations in the functions of language.

Hasegawa, D., Kaneko, N., Shirakawa, S., Sakuta, H., and Sumi, K. (2018). Evaluation of speech-to-gesture generation using bi-directional lstm network. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents* (ACM), 79–86

He, Y., Nagels, A., Schlesewsky, M., and Straube, B. (2018). The role of gamma oscillations during integration of metaphoric gestures and abstract speech. *Frontiers in psychology* 9, 1348

Iacobacci, I., Pilehvar, M. T., and Navigli, R. (2016). Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 897–907

Ishii, R., Katayama, T., Higashinaka, R., and Tomita, J. (2018). Generating body motions using spoken language in dialogue. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents* (ACM), 87–92

Iverson, J. M. and Goldin-Meadow, S. (1998). Why people gesture when they speak. *Nature* 396, 228–228

Johnson, M. (2013). *The body in the mind: The bodily basis of meaning, imagination, and reason* (University of Chicago Press)

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (USA: Prentice Hall PTR), 1st edn.

Kendon, A. (1980). Gesticulation and speech: Two aspects of the. *The relationship of verbal and nonverbal communication* , 207

Kendon, A. (1988). How gestures can become like words. In *This paper is a revision of a paper presented to the American Anthropological Association, Chicago, Dec 1983.* (Hogrefe & Huber Publishers)

Kendon, A. (2004). *Gesture: Visible action as utterance* (Cambridge University Press)

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*

Kipp, M., Neff, M., and Albrecht, I. (2007). An annotation scheme for conversational gestures: how to economically capture timing and form. *Language Resources and Evaluation* 41, 325–339

Krahmer, E. and Swerts, M. (2004). More about brows. In *From Brows till Trust: Evaluating Embodied Conversational Agents*, eds. Z. Ruttkay and C. Pelachaud (Kluwer)

Krahmer, E. and Swerts, M. (2007). The effects of visual beats on prosodic prominence: Acoustic analyses, auditory perception and visual perception. *Journal of memory and language* 57, 396–414

Kucherenko, T., Hasegawa, D., Henter, G. E., Kaneko, N., and Kjellström, H. (2019). Analyzing input and output representations for speech-driven gesture generation. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*. 97–104

Kucherenko, T., Jonell, P., van Waveren, S., Henter, G. E., Alexandersson, S., Leite, I., et al. (2020). Gesticulator: A framework for semantically-aware speech-driven gesture generation. In *Proceedings of the 2020 International Conference on Multimodal Interaction*. 242–250

Lakoff, G. and Johnson, M. (1980). Conceptual metaphor in everyday language. *The journal of Philosophy* 77, 453–486

Lee, J. and Marsella, S. (2006). Nonverbal behavior generator for embodied conversational agents. In *International Workshop on Intelligent Virtual Agents* (Springer), 243–255

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*. 24–26

Levine, S., Krähenbühl, P., Thrun, S., and Koltun, V. (2010). Gesture controllers. In *ACM Transactions on Graphics (TOG)* (ACM), vol. 29, 124

Levine, Y., Lenz, B., Dagan, O., Ram, O., Padnos, D., Sharir, O., et al. (2020). Sensebert: Driving some sense into bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4656–4667

Levitan, R. and Hirschberg, J. (2011). Measuring acoustic-prosodic entrainment with respect to multiple levels and dimensions. In *Twelfth Annual Conference of the International Speech Communication Association*

Lhommet, M. and Marsella, S. (2014). Metaphoric gestures: towards grounded mental spaces. In *International Conference on Intelligent Virtual Agents* (Springer), 264–274

Lhommet, M. and Marsella, S. C. (2013). Gesture with meaning. In *International Workshop on Intelligent Virtual Agents* (Springer), 303–312

Lin, A. S., Wu, L., Corona, R., Tai, K., Huang, Q., and Mooney, R. J. (2018). Generating animated videos of human activities from natural language descriptions. *Learning* 2018, 1

Loehr, D. P. (2012). Temporal, structural, and pragmatic synchrony between intonation and gesture. *Laboratory Phonology* 3, 71–89

Lücking, A., Mehler, A., Walther, D., Mauri, M., and Kurfürst, D. (2016). Finding recurrent features of image schema gestures: the figure corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 1426–1431

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60

McClave, E. Z. (2000). Linguistic functions of head movements in the context of speech. *Journal of pragmatics* 32, 855–878

McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*

McNeill, D. (1992). *Hand and mind: What gestures reveal about thought* (University of Chicago press)

Mehler, A., Lücking, A., and Abrami, G. (2015). Wikinect: Image schemata as a basis of gestural writing for kinetic museum wikis. *Universal Access in the Information Society* 14, 333–349

Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Elgharib, M., Fua, P., et al. (2020). Xnect: Real-time multi-person 3d motion capture with a single rgb camera. *ACM Transactions on Graphics (TOG)* 39, 82–1

Melamud, O., Goldberger, J., and Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*. 51–61

Menenti, L., Garrod, S., and Pickering, M. (2012). Toward a neural basis of interactive alignment in conversation. *Frontiers in Human Neuroscience* 6, 185. doi: 10.3389/fnhum.2012.00185

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119

Miles, L., Nind, L., and Macrae, C. (2010). Moving through time. *Psychological Science* 21, 222

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM* 38, 39–41

Miller, G. A., Chodorow, M., Landes, S., Leacock, C., and Thomas, R. G. (1994). Using a semantic concordance for sense identification. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*

Moro, A. and Navigli, R. (2015). Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*. 288–297

Moro, A., Raganato, A., and Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics* 2, 231–244

Navigli, R., Jurgens, D., and Vannella, D. (2013). Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. 222–231

Neff, M., Kipp, M., Albrecht, I., and Seidel, H.-P. (2008). Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Transactions on Graphics (TOG)* 27, 1–24

Nihei, F., Nakano, Y., Higashinaka, R., and Ishii, R. (2019). Determining iconic gesture forms based on entity image representation. In *2019 International Conference on Multimodal Interaction*. 419–425

Özçalışkan, Ş. and Goldin-Meadow, S. (2005). Gesture is at the cutting edge of early language development. *Cognition* 96, B101–B113

Pelachaud, C. (2009). Studies on gesture expressivity for a virtual agent. *Speech Communication* 51, 630–639

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543

Peshkov, K., Prévot, L., and Bertrand, R. (2013). Prosodic phrasing evaluation: measures and tools. *Proceedings of TRASP 2013*

Pouw, W., de Wit, J., Bögels, S., Rasenberg, M., Milivojevic, B., and Ozyurek, A. (2021). Semantically related gestures move alike: Towards a distributional semantics of gesture kinematics. In *Proceedings of the 23rd International Conference on Human-Computer Interaction*

Pradhan, S., Loper, E., Dligach, D., and Palmer, M. (2007). Semeval-2007 task-17: English lexical sample, srl and all words. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*. 87–92

Prillwitz, S., Leven, R., Zienert, H., Hanke, T., and Henning, J. (1989). Hamburg notation system for sign languages: an introductory guide, hamnosys version 2.0. *Signum, Seedorf, Germany*

Raganato, A., Camacho-Collados, J., and Navigli, R. (2017). Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. 99–110

Ravenet, B., Clavel, C., and Pelachaud, C. (2018a). Automatic nonverbal behavior generation from image schemas. In *Proceedings of the 17th international conference on autonomous agents and multiagent systems*. 1667–1674

Ravenet, B., Pelachaud, C., Clavel, C., and Marsella, S. (2018b). Automating the production of communicative gestures in embodied characters. *Frontiers in psychology* 9, 1144

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (Springer), 234–241

Rothe, S. and Schütze, H. (2015). Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1793–1803

Saint-Amand, K. (2018). Gest-is: Multi-lingual corpus of gesture and information structure. *Unpublished Report*

Saund, C., Roth, M., Chollet, M., and Marsella, S. (2019). Multiple metaphors in metaphoric gesturing. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)* (IEEE), 524–530

Scherer, S., Kane, J., Gobl, C., and Schwenker, F. (2013). Investigating fuzzy-input fuzzy-output support vector machines for robust voice quality classification. *Computer Speech & Language* 27, 263–287

Sekine, K. and Rose, M. L. (2013). The relationship of aphasia type and gesture production in people with aphasia

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*

Snyder, B. and Palmer, M. (2004). The english all-words task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. 41–43

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112

Swerts, M. and Krahmer, E. (2010). Visual prosody of newsreaders: Effects of information structure, emotional content and intended audience on facial expressions. *Journal of Phonetics* 38, 197–206

Taghipour, K. and Ng, H. T. (2015a). One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the nineteenth conference on computational natural language learning*. 338–344

Taghipour, K. and Ng, H. T. (2015b). Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*. 314–323

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. 252–259

Wagner, P., Malisz, Z., and Kopp, S. (2014). Gesture and speech in interaction: An overview. *Speech Communication* 57, 209 – 232

Wei, C. Y. (2006). Not crazy, just talking on the phone: Gestures and mobile phone conversations. In *2006 IEEE International Professional Communication Conference* (IEEE), 299–307

Wessler, J. and Hansen, J. (2017). Temporal closeness promotes imitation of meaningful gestures in face-to-face communication. *Journal of Nonverbal Behavior* 41, 415–431

Wilson, A. D. and Golonka, S. (2013). Embodied cognition is not what you think it is. *Frontiers in psychology* 4, 58

Wu, B., Liu, C., Ishi, C. T., and Ishiguro, H. (2021). Modeling the conditional distribution of co-speech upper body gesture jointly using conditional-gan and unrolled-gan. *Electronics* 10, 228

Yasinnik, Y., Renwick, M., and Shattuck-Hufnagel, S. (2004). The timing of speech-accompanying gestures with respect to prosody. In *Proceedings of the International Conference: From sound to sense* (Citeseer), vol. 50, 10–13

Zadeh, A., Chong Lim, Y., Baltrusaitis, T., and Morency, L.-P. (2017). Convolutional experts constrained local model for 3d facial landmark detection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2519–2528

Zhong, Z. and Ng, H. T. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 system demonstrations*. 78–83

Zhou, Y., Habermann, M., Xu, W., Habibie, I., Theobalt, C., and Xu, F. (2020). Monocular real-time hand shape and motion capture using multi-modal data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5346–5355