UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Chan Wai Tik

Emergent Theory of Mind (ToM) from Token Merging

Master's Thesis (30 ECTS)

Supervisor(s): Kallol Roy, Assistant Professor

Tartu 2023

Emergent Theory of Mind (ToM) from Token Merging

Abstract:

The large language models by using deep learning methods give surprising results in achieving human-level performance in a variety of tasks, especially on it shows the ability of Artificial General Intelligence(AGI) and the understanding of false belief from solely language training. This suggested a strong relationship between the Theory of Mind(ToM) and language training which has been a long study in the field of psychology. This rise the question on is it the same for those deep learning language models. In this direction, the thesis focus on the natural language understanding task(semantics classification) and gives a hypothesis that apart from the statistic, language model understanding is built on a deep structure from training. However, this structure is not directly accessible but only through tests, just like those studies of ToM in psychology. Thus, this thesis proposes a method - Token Merge that enables the test of the existence of the structure. The experiment result gives positive feedback on supporting the proposed hypothesis and it also gives an ordering of the importance of performance by grammatical tagging.

Keywords: Artificial Intelligence, Neural Networks, Large Language Model(LLM), Theory of Mind(ToM)

CERCS:P176-Artificial intelligence,H350-Linguistics

Emergent Theory of Mind (ToM) alates Tokenide ühendamine

Lühikokkuvõte:

Suured keelemudelid, kasutades süvaõppe meetodeid, annavad üllatavaid tulemusi inimtasemel jõudluse saavutamisel mitmesugustes ülesannetes, eriti see näitab tehisintellekti (AGI) võimekust ja valeuskumuste mõistmist ainult keeleõppest. See viitas tugevale seosele vaimuteooria (ToM) ja keeleõppe vahel, mida on psühholoogia valdkonnas kaua uuritud. See tõstatab küsimuse, kas see on sama ka nende süvaõppe keelemudelite puhul. Selles suunas keskendub lõputöö loomuliku keele mõistmise ülesandele (semantiline klassifikatsioon) ja esitab hüpoteesi, et peale statistika põhineb keelemudeli mõistmine koolitusest tulenevale süvastruktuurile. See struktuur ei ole aga otseselt juurdepääsetav, vaid ainult testide kaudu, nagu ka psühholoogia ToM-i uuringud. Seega on käesolevas lõputöös välja pakutud meetod – Token Merge, mis võimaldab testida struktuuri olemasolu. Eksperimendi tulemus annab positiivset tagasisidet väljapakutud hüpoteesi toetamise kohta ning annab ka järjestuse soorituse tähtsuse kohta grammatilise märgistamise teel.

Võtmesõnad:CERCS:P176-Artificial intelligence,H350-Linguistics

CERCS:P176-Artificial intelligence,H350-Linguistics

Contents

1 Introduction

The recent development in Large Language Models (LLM) such as ChatCPT has archived huge success and stormed the world with its capability of solving varieties of tasks solely based on mastery in language training[BCE$^+$]. However, the underlining mechanism is still unknown and hard to investigate due to the large parameter space. On the other hand, some research[Kos] suggested that LLM has developed a sense of the Theory of Mind (ToM) along the way during its language training as a by-product. Those researches drive the research focus of this thesis on how a language model consolidates information from words to become understanding.

In linguistics study, the hypothesis proposed by Chomsky[HCF02] suggested there are two senses of faculty of language: internal(I) and external(E) language(details in 1.1), and the primary study for linguistics is focused on I-language(though, mind) by modeling from E-language, and this results in the theory of generative grammar and universal grammar. In a high-level overview, the I-language described by Chomsky can be translated into the Theory of Ming(ToM) from psychology, but it is more well defined from the psychology aspect(details in 1.2) and a large amount of research shows there is a strong correlation between ToM and external language training. However, this type of influence is not captured by the linguistics field. With all those components described above, this thesis proposes a hypothesis that during language training, apart from statistics, a potential deep structure is formed which plays an important role. However, this structure is not directly accessible, thus a new method - token merge is proposed in order to exterminate the existence of such structure.

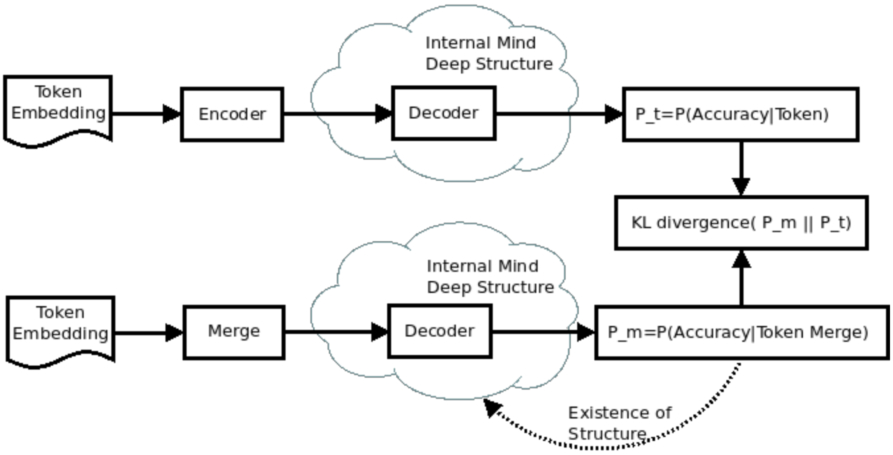

Token Embedding
Encoder
Internal Mind Deep Structure
Decoder
P_t=P(Accuracy|Token)
KL divergence(P_m || P_t)
Token Embedding
Merge
Internal Mind Deep Structure
Decoder
P_m=P(Accuracy|Token Merge)
Existence of Structure

Figure 1. Hypothesis on testing the existence of deep structure by token merge

The above figure(1) visualized the hypothesis that tries to test in this thesis. The test accuracy distribution will be used as a medium to see if it is significantly different between

3 cases: base case, always wrong, and token merge, those methods are summarized in the following:

1. Always wrong - destroy the conditional probability $P(token|target)$ by disturb training
2. Token merge - destroy the language structure in the input text
3. language model without any interference

The paper is organized into 5 sections, the first section is an introduction which will give some overview of the background, followed by two sections of literature review. The experiment details will be described in section four and followed by the result discussion and conclusion.

1.1 Language

Language has variate meanings according to different content and setting, as pointed out in[HCF02], DNA is a universal language that encodes biological information that is shared along all living organisms on earth, but not the case for communication. In the context of communication, human language also got significantly different from other species in its power of expression which is deeply correlated with the property of human language on hierarchy structured(grammar), generative, and most importantly recursive[HCF02]. The study of language(Linguistics) is integrated with a wide range of scientific areas from mathematics, philosophy, neuroscience, etc[GO22]. In viewing language from a biological point of view, bio-linguistics treats it as an evolving organism and reshape the study as the internal and external language (I-Language and E-Language)[GO22]. I-language is defined to be a mental mind which is intentional, internal, and individual and represents the computational aspect of language while the E-language is the observable language that humans use in communication[GO22, HCF02]. In this formulation, the study of linguistics focuses on inferring the mechanism of I-language by observing external language generated by internal language and formulating the grammar in daily use. The relationship between grammar, I-language, and E-language can be visualized below (fig:2) [MN06]:

One observation of the external language is its property on an infinite set of expressions together with the constraint on limited memory of the brain which suggested that I-language holds an abstract and generative structure behind the scenes and plays a crucial role in determining the interpretation[TNO+19]. From those observations on external language, Chomsky has proposed universal grammar (UG) and minimalist program(MP) to describe and characterize the formulation and property of I-language.
UG hypothesized that human language shares a certain degree of fundamental similarities such as general constraints on grammar and common property on features like lexical

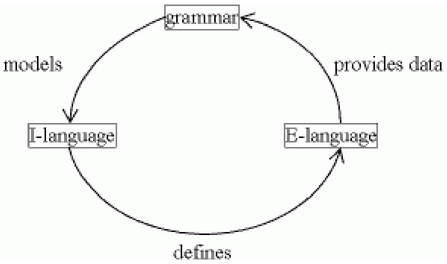


Figure 2. relationship between Grammar, I-language and E-language.

categories[Dąb15], etc., and this unique grammar follows the three-factor of language design[Cho05]:

- language-independent principles of data processing
- structural architecture
- computational efficiency

Those principles above assert the base of MP. In the framework of MP, it proposes a bare phrase structure(BPS) together with a simple operator - MERGE to construct it. The MERGE operator is very simple and takes 2 syntactic arguments and combines them to form a new syntactic object:

$$\text{MERGE}(\alpha,\beta) = K,\ \alpha,\beta$$

Here α and β are the two arguments and K is the new object often called "label". The most difficult part here is how to determine the label of the new object, detail can be found in [1] Here is a visualization example of BPS and MERGE operation with lexical categories (fig:3): However, this type of structure can generate syntactically correct but semantically meaningless language as shown by Chomsky in his book " Syntactic Structures".

[1] LIN331 – Syntactic Theory- https://nlacara.github.io/teaching/331S18/331-7-bps.pdf

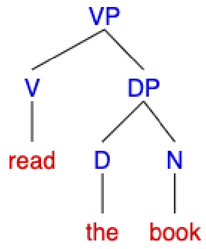


Figure 3. Example on BPS and MERGE (source wiki https en.wikipedia.orgwikiMerge$_{\%}28linguistics\%29$

1.2 Theory of Mind (ToM)

Theory of Mind has a long history and is an active research field along a broad range of science like developmental or social psychology, etc, and neuroscience. In the simplest explanation, it can be referred to as the ability of humankind to infer either other's or self's feeling, beliefs, and thoughts which summarize as mental state[BM13]. However, the term mental state is hard to define and quantify due to its complex and abstract concept and the difference in focus on research with respect to variate fields, but it is generally accepted that the ToM concept is a composition of cognitive skills to understand the belief and emotions[BLGB20], and base on that, experiment and test on cognitive skills can be developed into 3 categories which summarized in the following table(1)[BLGB20, BM13]:

Mental State	Cognitive Skill	Experiment/ Test
Belief	Shared world knowledge	• Text-based tasks • Non-verbal picture-based tasks
	Interpreting actions	• False belief tasks
Emotion	Perceiving social cues	• Facial/Vocal emotion recognition

Table 1. Mental State and Related Cognitive Tasks

The above table shows that to access the mental state of belief, one can test the target with either context or false belief base understanding. In the context base test, participants are usually given a textual or picture set of social scenarios, and questions are given related to characters of that scenario either on classification(what the character thinks of) or predication(guessing action taken by the character) to access the inferred mental state of participants. However, there are a number of considerations that need to be careful of as those test demand heavily on working memory and level of language skill.

The false-belief test is one step further based on the above tasks which include differences in the mental state of participants and the character in the story[BM13]. In this type of experiment, participants are usually given a storyboard that shows a particular event that alters the result but is only known to the participants but not the protagonist of the story. In this way, the participant is asked to predict the action base on the character's belief(false belief: lacking knowledge of the event) and the belief of the participant's mental state(true belief). This test provides a deviation from the reasoning that humans experience daily which is based on true belief[BM13]. An example of a false belief story is provided in fig(4) for reference. Social cues are another important phenomenon to observe mental state changes. Those cues are usually referred to as gaze, facial expression, vocal changes, etccite. While gaze cues are more focused on attention, facial/vocal recognition is related to determining the emotional state of others.

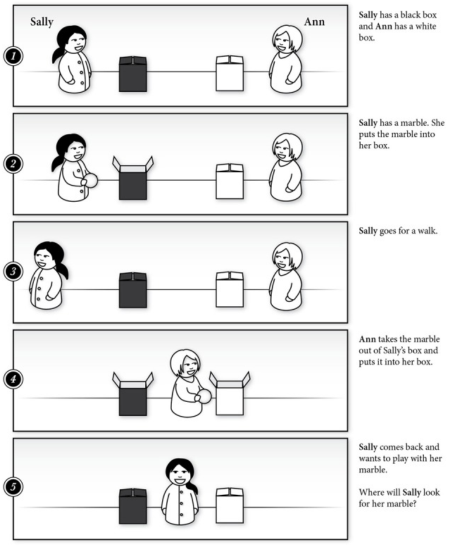

Sally
Ann
1
Sally has a black box and Ann has a white box.
2
Sally has a marble. She puts the marble into her box.
3
Sally goes for a walk.
4
Ann takes the marble out of Sally's box and puts it into her box.
5
Sally comes back and wants to play with her marble.
Where will Sally look for her marble?

Figure 4. False Belief Example [Wim83]

1.3 Language Model

Since the spark of linguistics, researchers try to model language in a scientific fashion by structural (as described in 1.1) or statistical analysis. In the early day of linguistics which solely relies on hard-crafted rules parsing that was purely symbolic[NOMC11]. When the time came to the digital age, the use of computers largely influenced research, especially in the power to handle huge textual data and the use of machine learning methods which give rise to natural language processing(NLP) for using computers to process human language.

In the domain of NLP, it can be subdivided into 2 main categories, natural language understanding(NLU) and natural language generation(NLG). In the simplest form, NLU can be understood as it performs syntactic and semantic analysis based on the contents in order to classify the meaning of the given text input. NLG on the other hand, is the process of generating text that meets some specific requirement with a given input and condition, for example, text translation.

In order to handle textual data before modeling(NLU or NLG), the data have to go through several numbers of pre-processing before feeding to any models. Those steps can be summarized in the following[DSV15]:

- Tokenization
- Stop word removal
- Stemming / Lemmatization -
- Embedding -

1.3.1 Tokenization

In English Textual data, it usually comes in paragraph format and needs to break down into atomic levels which can be done down to individual character levels, but the usual way will be chopping into words as tokens. However, there are problems arise in chopping at the word level which is mostly related to short-form writing style and punctuation, for example: isn't, O'Neil, while it is relatively easy to handle with the case on O'Neil(change to O Neil), short-form need to be handle in care. It is also reminded that tokenization is language specific where the strategy change with respect to language.

1.3.2 Stop word Removal

The tokenization process will generate a huge amount of tokens from the text data, but some of the tokens occur very frequently which may not contribute any meaning to the text itself. Removing those tokens will help in reducing the dimension of word space. There are several methods to determine stop words, for example, pre-defined list, term frequency, inverse document frequency, etc.

1.3.3 Stemming/Lemmatization

English words can exist in different forms like initial, initialed, initialing, and initialization which all refer to the root initial. Stemming is the process to identify the root of the token and aims to reduce the inflectional form in order to reduce the size of tokens. The purpose of lemmatization is the same, but with different methodologies which use vocabulary and morphological analysis of words to apply the transformation while Stemming is more rely on a heuristic process to truncate part of the word.

1.3.4 Embedding

Tokens are still in textual format after those processes, however, most language models work on numerical space rather than discrete text space, the embedding process is to transform tokens into numerical form as a final step before input to the model. There are varieties of ways to achieve the purpose, the simplest one will be one-hot encoding which turns each token into a binary array with length equal to the size of tokens. The disadvantage of this is it creates a very high dimension and sparsity matrix to represent tokens which may impact the model's convergence time and performance. Other's methods include bag of words together with term frequency-inverse document frequency(TF-IDF) to assign a numerical value for tokens. However, the trend changed to use Word2Vec after adopting the neural network which uses the vector representation before the output layer as the token's embedding. The task of the neural network can be either using surrounding context(words) to predict the target word or the opposite way. This way, it heavily reduces the embedding dimension in comparison to the one hot-encoding and uses the surrounding context to assign numerical meaning to tokens.

1.3.5 Language Model with Deep learning

During the pre-deep learning era, there is a wide range of modeling such as support vector machine(SVM) that is based on statistical aspects, but the accuracy heavily depends on the data and the prior belief on the kernel selection. After the deep learning era, especially after the invention of the Long-short term memory model (LSTM, an expansion of recurrent neural network RNN) and Transformer (BERT) which change the trend from statistical to auto-regressive model. The main difference between the two is that RNN-type models are equipped with state space learning[Sie95] to enrich the representation of data while the transformer relies on an attention mechanism to build hidden representation from the input text.

RNN has a long and successful history in sequential modeling before the invention of the transformer. The main distinguishing property of RNN from others is its feedback loop that enables influences of the future by information aggregation from the past and current. The architecture of RNN is shown in the following diagram(5 source:[2])

[2] https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg

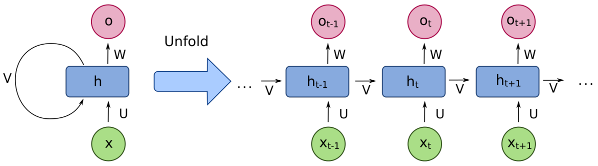


Figure 5. RNN structure, Left-hand side - Fold version

The LSTM is a further development base on the RNN structure which comes with the ability on decided what to keep and forget from the input and past information by using control gates(input/forget gate). However, it is prompt that LSTM suffers from the problem of handling long-term dependence which it forgets previously seen data if the input sequence is longcite. That creates a bottleneck for LSTM on textual modeling since text dependence can be across multiple lines or paragraphs.

The situation has changed after the publication of the paper "Attention is all you need"cite. The purposed attention mechanism totally throws away the recurrent network and later becomes the transformer model with multiple-head attention and a deep encoder network. The attention formulation can be briefly described as query retrieval with the query representing the current input token and the key being the reset of the corpus that returns the similarity between the pairs of words, in this sense, it provides a view of global dependence which against the problem of long-term dependence within the textual context. This attention block is further stacked up vertically(Multi-Head) and horizontally(composition transformation) to form the base of all transformer networks.

2 ToM and Language - Literature Survey

From the introduction, it is shown that language(I-language) and ToM is strongly coupled in Linguistics and psychology but has different aspects of external language. Linguistics model E-language as data generated from internal, treating internal language as deep structural(or grammar) that is universal to everyone, but psychology is more focused on how **external language is affecting the development of the mental state**. There is no doubt that language does matter in mental thought, but the question is what's **the role of language in building up cognitive structures and their uniqueness in that role[JWA05].** The investigation of the roles of language in ToM can be summarized into four main focuses[JWA05]:

1. No role at all

2. Conversational Pragmatics

3. Lexical Semantics
4. Complementation Syntax
5. Synergy (Combine action)

While the no role at all suggested that language is only a tool to access and implement the abstract human mind, others give support that language is crucial to the development of ToM in the form of interaction between the surrounding environment during the infant and pre-school age. All those roles for language are conversational based and each of them provides an explanation of how language helps in build-up ToM, for example, conversational pragmatics suggests language as information exchange promotes the understanding of there are different thoughts by others on the same fact, while lexical semantics and complementation syntax are emphasizing the role of language in formulations of abstract concept from lexical mental verb(e.g happy, sad) and sentential complement.

While the above paragraph describes the relationship between language and ToM in a constructive way, much developmental psychology research provides both correlation and training study in order to show the importance of language. Those studies are primarily focusing on false-beliefs understanding ability as it reveals both internal and external beliefs at the same time. In correlation studies, different age groups of children are examined on both false-belief tasks and language testing to identify the relationship between language capability and the performance of ToM[Ebe20]. On the other hand, training study is done by giving variate language training to children and looking for performance differences given by the training. Those training usually variate in the use of words, for example, with or without restriction on the usage of mental verbs (e.g. think, know) and absence of sentential complement sentences[LT03]. One of the research[PS09] takes a step further to investigate language in other forms Sign language that is used by deaf people which results in the same conclusion.

The research above suggests a strong correlation and the importance of language in the role of ToM development, especially when there is defective during language acquisition, it heavily affects the mental state of understanding in all forms of language.

3 Language Model - Literature Survey

The rise of ChatGPT is storming the world over the past year and it still influencing the world heavily in a wide range of areas. On the research aspect, there is no solid theory that can explain the behavior of those LLM, especially on the generality across different domains of knowledge by simple language training[BCE$^+$]. The study on[BCE$^+$] gives a details study on the intelligent ability of LLM and the research on [Kos] shows that GPT3.5 solved 90% of false belief tasks that reached the performance of a seven years old child, it further hypothesized the result of the large language model is attributed to the internal development of ToM as a by-product from solely language training. This hypothesis matches perfectly with the above discussion on the relationship between language and ToM. The basic architecture nowadays for language models is dominated by the transformer which composite of an autoencoder and attention mechanism together, without the recurrent unit involved[GG20]. However, the positional information is important in language, it is pushed into the embedding as an extracted feature of tokens to represent the sequential order within the text. The main idea of the transformer is to encode the information from input text by the composition of transformation(attention) which result in a complex hidden space for the decoder to perform the relevant tasks such as predicting a missing word in a sentence or the next word generation[GG20]. Since the composition of attention is formed recursively by pairs of pair of words, it forms a hierarchical structure of measurement that reproduce the property of maximum mean discrepancy used in measuring the difference between distribution from their samples[GG20]and it usually comes with several attention stacks together to form multi-head attention, which it can be thought of capturing different statistical property from the input data. But the question of the number of attention needed and the efficiency of increasing attention remains unanswered[Mer]. It is shown that transformer architecture mainly benefits from the attention mechanism which has its own place in statistics but is this statistical property gives rise to such high accuracy or even the enlightenment of ToM still remains unclear.

4 Experiment Setting

4.1 Data Set and Pre-processing

The experiment is conducted on the IMDB movie review data set cite for sentiment classification to mimic inferring mental states(ie. good or bad) from the human text. The whole data set consists of 50,000 textual reviews and a target label indicating positive and negative feedback on the movie. 15,000 reviews are sampled from the data as the training set and 3,000 for testing data. The following table(2) summarizes the positive and negative class distribution on both the training and test set.

Train/Test	Number of Review	Percentage(Pos / Neg)
Train	15000	50% / 50%
Test	3000	50% / 50%

Table 2. Training, Testing Set

Those data sets then go through data cleaning(remove html related tags) and pre-processing pipeline as described in 1.3 apart from the last two processes (Stemming/Lemmatization, embedding). The average token count for each review is 105 (=1,581,851/15,000).

Since the experiment is conducted on destroying the language, the part of speech(POS) tags are used as metadata to control which part of the text should work on instead of doing it at individual token level. The following table(3) shows the top 10 POS tags count and the corresponding percentage for the pre-processed data set.

Data Set	POS tag	Positive review	Negative Review
Train	NOUN	352,846(22.3%)	336,466(21.2%)
	VERB	165,504(10.4%)	167,988(10.6%)
	ADJ	140,768(8.8%)	133,559(8.4%)
	PROPN	86,150(5.4%)	60,792(3.8%)
	ADV	37,888(2.3%)	40,703(2.5%)
	NUM	10,036(0.6%)	10,977(0.6%)
	ADP	4,911	5,620
	DET	2,429	2,300
	INTJ	2,021	3,435
	X	2,008	2,292
Test	NOUN	74,954	69,862
	VERB	32,464	32,254
	ADJ	31,073	29,371
	PROPN	16,473	11,732
	ADV	7,580	7,715
	DET	2,069	2,005
	NUM	1,870	2,073
	ADP	937	1,146
	INTJ	359	635
	X	344	433

Table 3. POS tag count

4.2 Simulate Incomplete Language Learning

The experiment result from developmental psychology suggested a strong correlation between deficiency in language and ToM. The simplest way to simulate the lack of certain types of words in the language is to skip those in the data set, however, it is not appropriate as those psychology experiments avoid using some type of words, but it didn't change the underline meaning of the sentence. In order to reflect this scenario in the language model, this experiment proposes 2 methods that try to approximate language deficiency, "always wrong" and Merge.

4.2.1 Always Wrong

In the Bayesian inference setting for semantic classification, the relationship between the target and token is usually formulated as:

$$P(target|tokens) = P(tokens|target)(target)$$

In simple words, the posterior probability is modeled by the influence of data and prior probability. Since the prior distribution is different by case, the only way to destroy the posterior is through the conditional probability part($P(tokens|target)$), which is the main idea behind the method "always wrong". In order to trick the model's classification at a specific position of the input text, it utilized the LSTM's output layer property that each output is corresponding to the input token position, so that it is possible to have a one-to-one mapping between the result and input token. In this experiment, the position is selected using POS tag, for example, treat all predictions on input token with ADJ tag as wrong. The illusion is done by changing the loss value at the corresponding location to 0.5 for the cross-entropy loss function. Figure(6) shows briefly how the method works, detail of the full model can be found in 4.3.

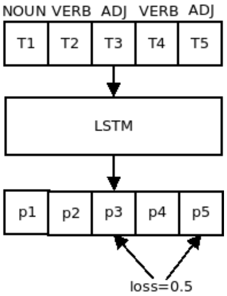


Figure 6. Trick the model on ADJ

4.2.2 Token Merge

The concept of merge tokens origin from the MERGE operator(1.1) of bare phrase structures, as introduced above, MERGE help to build a structural description set from sentences and assume to take 2 input. In this experiment, 3 strategies are used to simulate the MERGE:

1. Random Merge
2. Merge surrounding words of a specific position
3. Merge specific POS pattern

Again, the specific position in the second strategy is relying on POS tag, for example, merge tokens that must include ADJ, on the other hand, the third way is to merge tokens in a specific ordered pattern such as ADJ+NOUN. Since it is known that there are common phrases that use in the English language grammar(ex. NOUN phrase), the merge action here is aimed to destroy that structure before it feeds to the model. The details of the merge will be described in 4.3.1, and the figure(7) below shows an example for each strategy.

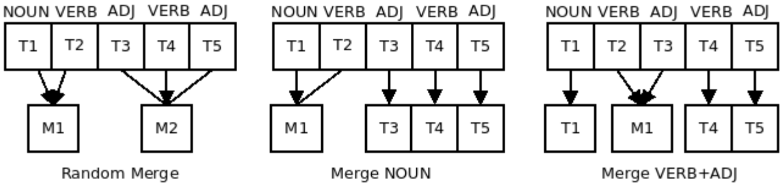


Figure 7. 3 ways to Merge

4.3 Model Architecture and Training

The architecture for the model consists of 4 main blocks which are Embedding, Merge, LSTM, and classification. The following table(4) and diagram(8) give a summary of the parameters set for each block and visualization for the model:

Block	Parameters
Embedding	embedding dim=256
Merge(Conv1D x2)	kernel=2,stride=1
LSTM	input dim=256, hidden dim=256, hidden layer=3, bidirection=True
LSTM Input sequence	5,10,25,50,600
Classification(Linear)	input dim=256*2 , out dim = 3

Table 4. Model Settings

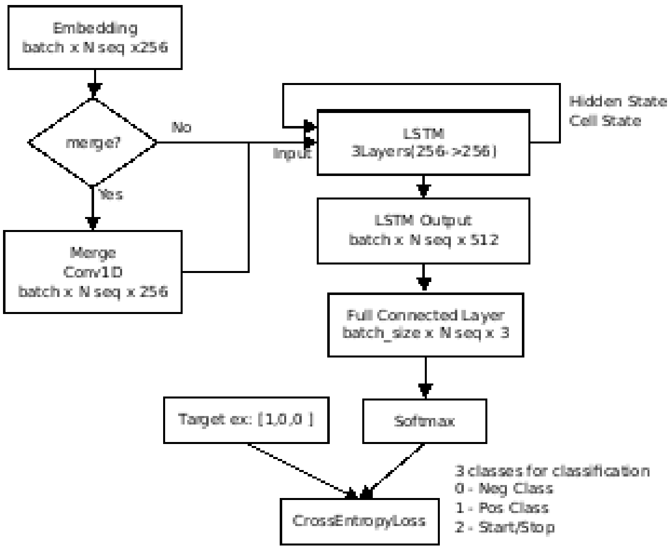


Figure 8. Full model Architecture

4.3.1 Token Merge Block

As described in 4.2.2, one of the methods to simulate incomplete language is by merging specific tokens which is done by function prepareMerge(1) and Conv1D. There are 3 parameters to define for merging:

1. α the minimum token to merge
2. β the maximum token to merge
3. l a random merge list of size equal input sequence length, lower and upper clipped by α, β (ex: [2,3,3,2,3,2])

After tokens pass through the embedding layer, it will result in an embedding tensor($\mathbb{E}$) of size NxE (N=sequence length, E=embedding dim). This matrix will then transform to a merge tensor(M) of size NxExβ by selecting the number of tokens to merge($l[i]$) with padding zero at the back if the number of tokens is smaller than β. The following code demonstrated how the merge is prepared before feeding to Conv1D.

The merge tensor $\mathbb{M}$ will be used as input to Conv1D(k=2,stride=1) so that each slice of $\mathbb{M}$ is transformed to the size of (E,1). The following figure(9) visualized the size transformation from embedding until merge:

Algorithm 1: PrepareMerge

```
//E=embedding dim,N=sequence length
Data: $l = [2, 3, 3, 2, 3, 2]$
Data: $\mathbb{E} = Embedding(tokens).reshape(E, -1)$
Data: $\mathbb{M} = tensor.zero(N, E, \beta)$
//Start from the end
$y \leftarrow l.length() - 1$
$c \leftarrow l.length()$
while $c \neq 0$ do
    $\mathbb{T} = tensor.zero(E, \beta)$
    $m \leftarrow l[y]$
    $s \leftarrow (c - m)$
    $\mathbb{T} = \mathbb{E}[:, s : c]$
    $\mathbb{M}[y] = \mathbb{T}$
    $c \leftarrow c - s$
    $y \leftarrow y - 1$
```

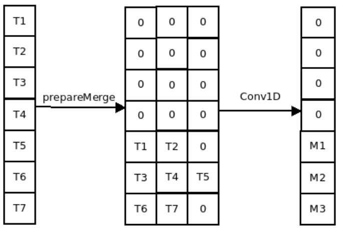


Figure 9. prepareMerge: N=7, $l = [2, 2, 3, 3, 2, 3, 2]$

4.3.2 Training

The model is fed with review by streaming a sequence of blocks, and the block size and the sliding windows are set to those pre-defined LSTM input sequence lengths. At the beginning of each review, the hidden and cell state of LSTM is initialized to 0 and both states of the current block are then fed back to the model together with the next block of input(fig.10). So, the average prediction made for one review is around 105 times. A mini-batch of size 1000 reviews is sampled from the training set as 1 epoch of training, then the test set is fed after each epoch for evaluation. The performance of the model is measured by the classification made on the last 20 tokens ($\hat{\mu}$), the result is treated as correctly classified if:
$mean(abs(\hat{\mu} - targe)) < 0.05$ (ie. at most 1 token prediction is wrong)

The following tables(5,6) summarized the general configuration and the incomplete language training setting for experiment runs.

sequence length	mini-batch size	Optimizator	Loss function
5,10,25,50,600	1000	Adam(default parameters)	CrossEntropy

Table 5. General Settings

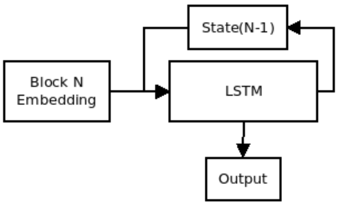


Figure 10. Feed Back

Incomplete language Method	POS tag
Always Wrong	NOUN, VERB, ADJ, ADJ+NOUN, NOUN+VERB+ADJ
Random Merge	N/A
Merge Specific Tags	NOUN,PROPN,VERB,ADJ,SCONJ,CONJ
Merge Specific Patten	ADJ+NOUN, NOUN+VERB+ADJ, NOUN+VERB

Table 6. Incomplete Language Setting

5 Experiment Result and Discussion

In this section, the experiment results of the two incomplete languages setup are presented on the test set. First of all, the base case results for each sequence length are shown in the following table(7).

	Test set Accuracy(%)				
sequence length	mean	quartile 25%	median	quartile 75%	IQR
5	0.66	0.656	0.668	0.679	0.023
10	0.681	0.683	0.701	0.713	0.03
25	0.708	0.730	0.740	0.745	0.015
50	0.762	0.773	0.778	0.781	0.008
600	0.754	0.768	0.792	0.801	0.033

Table 7. Base Case Result (450 epochs for 2 runs)

The above base case results show that the mean accuracy is growing together with increasing sequence length of blocks. One thing to notice is that apart from the sequence length of 5, all others' accuracy distribution is left-skewed. The mean accuracy of each sequence length will be used as a baseline to compare the experiment result on those deficiency language training, so, results from onward are **changed to Ratio Based where the divisor is the mean accuracy of each sequence.** Thus, the interruption of the value from each table below is the **performance degradation relative to the base case** with 1 representing no changes and 0 means fully degraded.

5.0.1 Always Wrong Method

The result of "always wrong" is separated into 2 categories, the first one is to show the result on only 1 POS tag specified while the second one is using more than 1 POS tag. The performance of the model is listed in the following table(8), details of result can be found in (15,16)

seq len	Degrade 1	Degrade 2	Degrade 3	Average Degrade
5	1	0.97	0.95	0.97
10	0.96	0.96	0.94	0.95
25	0.96	0.98	0.98	0.97
50	0.95	0.94	0.97	0.95

Table 8. Performance Degrade by always wrong.
Column 1-(NOUN,VERB, ADJ)
Column 2-(ADJ & NOUN)
Column 3-Other's

In general, the above results show that the "always wrong" method doesn't largely impact the performance, **with an average degradation from 3% to 5%**. Especially, with the sequence length of 5, it shows nearly no difference in comparison to the base case. It is also stressed that those are from a disrupted training which on average over 50% of loss value is modified(table3). Although the mean accuracy looks good in this experiment, it shows some differences in the distribution once comparing the inter-quartile range(IQR), and the ratio between the base case shown in the following table(9).

seq len	(NOUN,VERB,ADJ)	(ADJ & NOUN)	Other's
5	1.6	1.4	1.4
10	0.7	1.13	1.13
25	1.26	2	1.9
50	3.7	5.3	1.7

Table 9. IQR comparison (Ratio)

Those figures suggested that for larger sequences of blocks (ie. >=25), the accuracy distribution is more affected by the method and the most contribution to the change in IQR is by using ADJ and NOUN together. A control case of always wrong with "NOUN & VERB & ADJ & PROPN & ADV & INTJ" is done and the average accuracy is 0.16.

5.0.2 Token Merge - Random Merge and Merge on specific Tag

The experiment results on token merge are presented in 2 parts, the first part will focus on the comparison between random merge and merge on POS tag, followed by the second part on presenting the result of merge with a specific pattern. The random merge is done with minimum and maximum merge set to 2 and 3 correspondingly. The result is summarized in the following(10):

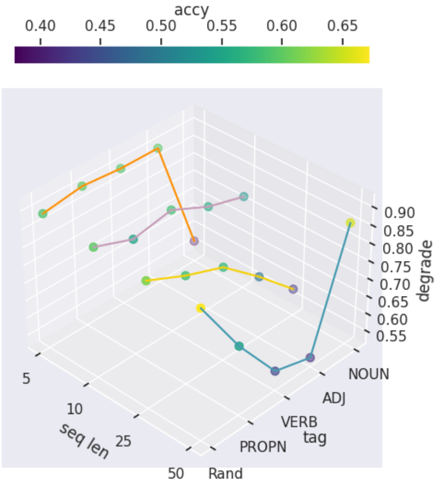
seq len	Random	PROPN	VERB	ADJ	NOUN
5	0.89	0.90	0.90	0.90	**0.573**
10	0.88	0.84	0.86	0.81	0.78
25	0.87	0.82	0.78	**0.69**	**0.59**
50	0.88	0.71	**0.57**	**0.54**	0.92

Table 10. Performance Degrade by Random Merge & POS Tag Merge.
Font in bold = accuracy< 0.5

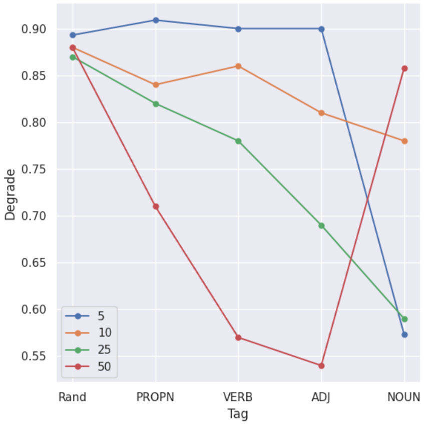
The result of random merge on all sequences length shows roughly the same level of degradation($\approx 12\%$). It also noted that in the random merge experiment, **the merged pattern is different** for short(≤ 10) and long sequences (≥ 25). The detail on pattern differences and test results are attached in the appendix(23,19,20). The above table shows a clear difference between short and long sequence lengths such that shorter sequences are more invariant to token merge, but not for longer sequences. The performance degradation is increasing according to the **tag ordering: PROPN, VERB, ADJ** for both 25 and 50 sequences, but not for NOUN. The following diagram(11a,11b) provided a better view of the relationship between degradation, tag, and sequence length.

Those diagrams show a few interesting observations on long sequences. First of all, the curvature of degradation is concave down for a sequence of 25 but concave up for 50 sequence length, this indicates that the drop in performance is more rapid with increasing sequences, this suggested that **structural importance is positively correlated with increasing sequence length**.

Furthermore, the ordering of degradation by POS tag is the same with ADJ getting the worst performance if excluding NOUN in consideration, however, the merge pattern count from the appendix(23) shows that ADJ occurs frequently as well on Random case, thus, it is curious on what casing the performance differences and it is done by studying the frequency count(table 11) on merge pattern for the two strategies From the table, the major difference is that ADJ+NOUN comes to the top with the strategy of ADJ, then follow by the top 1 and 2 patterns from the random merge. **This may suggest that ADJ+NOUN is significant for the degradation.** Another point to note is the pattern NOUN+ADJ+NOUN which may contribute to the degradation, but the count is just slightly over 10%. However, there is no cue for the different behavior of merging with NOUN between the sequence length of 10,50, and 5,25. The ordering of tags by perfor-



(a) Token Merger: Degradation vs Sequence len vs Merge Tag



(b) Token Merger: Degradation vs Merge Tag

Top 10 Average Merge Pattern Count %			
Random		ADJ	
ADJ_ADJ_ADJ	0.214	ADJ_NOUN	0.235
ADJ_ADJ	0.184	ADJ_ADJ_ADJ	0.231
NOUN_NOUN	0.152	ADJ_ADJ	0.187
ADJ_NOUN	0.086	NOUN_ADJ_NOUN	0.108
NOUN_VERB	0.086	ADJ_ADJ_NOUN	0.071
VERB_NOUN	0.074	VERB_ADJ_NOUN	0.067
NOUN_NOUN_NOUN	0.067	ADJ_VERB	0.032
ADJ_NOUN_NOUN	0.047	NOUN_ADJ	0.029
NOUN_ADJ	0.047	ADV_ADJ_NOUN	0.021
NOUN_VERB_NOUN	0.044	DET_ADJ	0.020

Table 11. Top 10 Average Merge Pattern(Rand Vs ADJ)

mance degradation happens to match with the top 10 POS tags count(table 3) in reverse order which may suggest the result is just a reflection of POS tags distribution(high frequency ⇒ more difficult for merge to learn), however, if considered with on the merge count of different strategies(table 12), the merge count of ADJ is roughly the same with Rand but with a huge gap degradation.

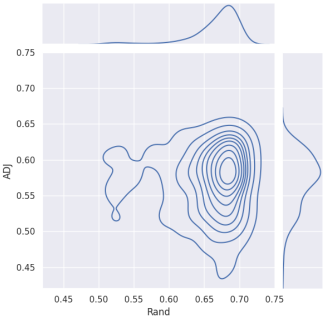
Apart from the measure of average degradation of performance, it is also wondered if those tests' accuracy has notably different from each other, so the Wilcoxon signed-rank test is used to test the median accuracy of ADJ is different from others, it is done on the result of 25 sequences length with a sampling data point of 290. The test shows support for the alternative hypothesis that the accuracy median is different between VERB and ADJ with p-value 2.05×10^{-11} at 95% confidence level. On the other hand, the same

Average Epoch Merge Count

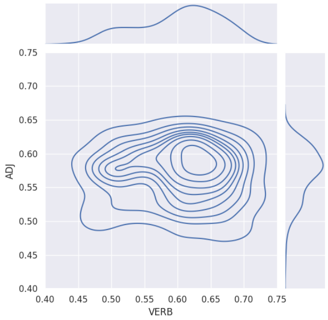
seq	Rand	PROPN	VERB	ADJ	NOUN
5	1	0.36	0.88	0.90	1.69
10	1	0.24	0.70	0.69	1.02
25	1	0.19	0.53	0.88	0.76
50	1	0.13	0.36	0.89	0.51

Table 12. Average Merge Count per epoch(Ratio to Random Merge)

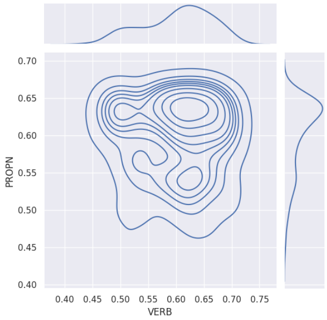
test is applied for VERB and PROPN, the result shows strong evidence to support the null hypothesis that their median is equal(p-value 0.7). The details of the statistical test can be found in the appendix(24). The following figures show the accuracy distribution for each case(12c):



(a) Join distribution plot on Accuracy
ADJ & Random



(b) Join distribution plot on Accuracy
ADJ & VERB



(c) Join distribution plot on Accuracy
VERB & PROPN

5.0.3 Token Merge - Merge with specific Pattern

The experiments result above suggested that the sequence block size will result in different behavior on the performance degradation which can mainly be separated into shorter(≤ 10) and longer(≥ 25), so, the fixed pattern merge testing will switch to focus on sequence length of 10 and 25 on different merge pattern. The patterns that are used are concerned with POS tags ADJ, VERB, and NOUN as a result of the previous section which gives insight into the pattern ADJ+NOUN may affect the underlying structure. The tables(13,14) and diagram(13a,13b)below give an overview of the degradation according to different patterns, and the details are placed in the appendix(25,27,26).

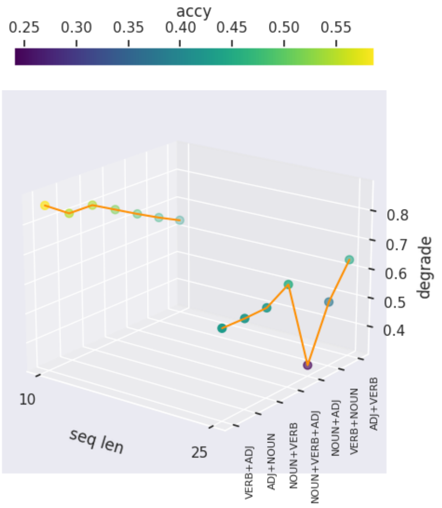
The result above indicates that the sequence length of 25 on all fixed pattern merges can only achieve $50\% \sim 60\%$ of the base case performance and the average accuracy are already under 0.5 which means it fails to learn in all case. In contrast, the short sequences still maintain over 70% performance of the base case on some patterns(VERB+ADJ,

seq len	Predicative Adjective VERB+ADJ	Attributteive Adjectives ADJ+NOUN	Postpositive Adjective NOUN+ADJ
10	0.862	0.807	**0.718**
25	**0.605**	**0.602**	**0.339**

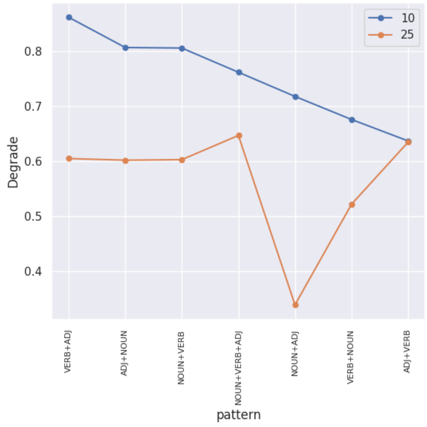
Table 13. Merge fixed pattern(1)

seq len	Subject NOUN+VERB	NOUNT+VERB+ADJ	ADJ+VERB	Object VERB+NOUN
10	0.806	0.762	**0.637**	**0.676**
25	**0.603**	**0.646**	**0.636**	**0.549**

Table 14. Merge fixed pattern(2)



(a) Fixed Pattern Merge: Degradation vs Sequence len vs Merge Tag



(b) Fixed Pattern Merge: Degradation vs Merge Tag

ADJ+NOUN, NOUN+VERB NOUN+ADJ+VERB), but also notice that the accuracy is just slightly over 0.5. In addition, the shorter sequence length shows the ability to infer on fixed pattern ADJ+NOUN but not for the long block which may suggest that the structure they acquired is different. However, it is found that the fixed pattern merge frequency count only got ($34\% \sim 6\%$) when compared with the random merge cases. It gives rise to the question of rather the training epoch need to be extended in order to draw the conclusion. (Those experiments were training with around 600 epochs and taking 3 days on average.)

6 Conclusion

The above experiments have demonstrated two different methods that try to degrade the performance of a small language model in semantics classification. The findings from those experiments can be summarized in the following 5 points:

1. Tricking the model by manipulating $P(token|target)$ won't harm the accuracy much, but it affects the distribution of accuracy.
2. Destroying language structure by token merge is much more harmful on the model to infer mental concepts (good/bad), thus giving support on language model do form structure from training.
3. Sequence length is negatively correlated with the model's accuracy on token merge.
4. Shorter sequence of input is invariant to structural changes with the price on accuracy.
5. There is an ordering on the model's performance with token merge with respect to POS tags.

Those findings also give rise to many questions from the statistical aspect. One may point out that the distribution of positive and negative tokens is unclear such that those POS tags in use are not right on the target. But the point of using POS tags as metadata is to avoid this problem. The idea is to treat the data as an observational study and use POS tags as an instrumental variable(IV) (fig14 upper path). In this way, the treatment will be normal training when it is off(not in the list of tags) and disturb training when it is on. Then, the true effect on treatment can be estimated from the accuracy. From the results above, it concluded that manipulating the token distribution by destroying the possible connection to the target by disturbing the learning has a small effect on the outcome. Furthermore, both positive and negative related tokens are grouped under part of speech, for example, "good" and "bad" are under adjectives tag, so that the disturb on learning is applied in a fair sense. The interpretation from IV suggested that the effect of tokens with respect to POS Tag partition has a small effect on the outcome and gives supporting evidence to the belief that the learning of semantics classification is not limited to statistical, but also structural. A visualization for the above idea is given in the figure(15).

Another thought from the statistical side is that the model still keeps counting on the occurrence of tokens since from the observation that the embedding weight doesn't change much from the initial(6), so it can still infer statistics property from the data. This question makes a strong argument for the existence of grammar structure constructed by the model, but if the argument holds, the merge performance won't be different much in comparison to the always wrong method. For instance, let's think of merging two

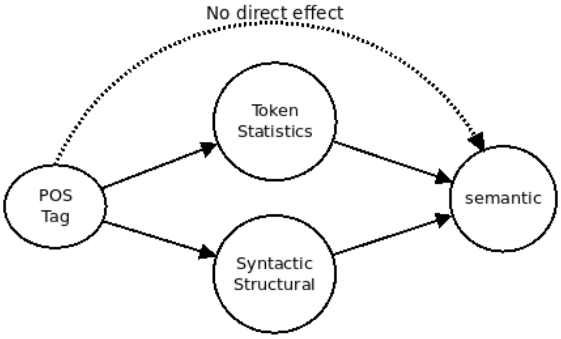


Figure 14. Instrumental Variable

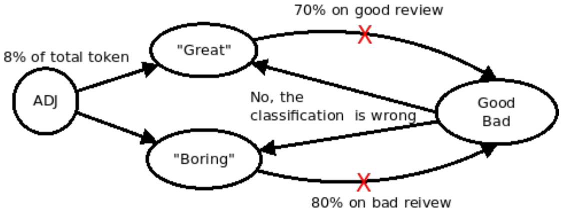


Figure 15. Disturb training to destroy the token statistic

words, "good" and "movie" into "gomodoive" by the Conv1D and it converges to this representation, then the model will still count this human non-sense word "gomodoive" to positive semantics, and not degrade much on the accuracy. The concept is illustrated in the following figure(16). The difference between always wrong and merge performance suggests that there is a structural building up by the model during the training process.

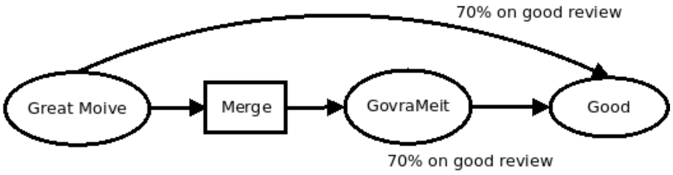
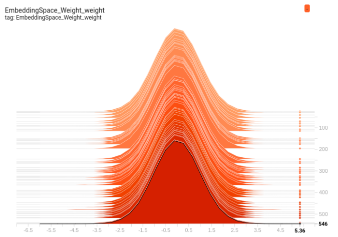
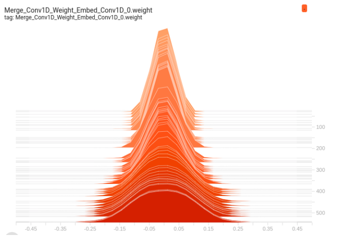


Figure 16. Statistics Point of view on Merge

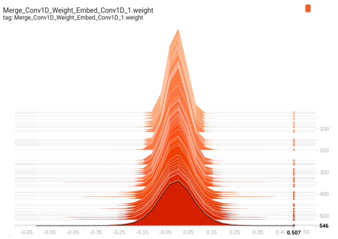
However, it may criticize that apart from the structure, the merge destroys the representation of the token and try to merge non-sense word together. It is admitted that this question is not easy to answer and requires more research and experiments, but yet, there are some findings that can be shared in order to prepare a deeper investigation. First of all, it is observed that the weight distribution of the embedding layer doesn't change much in both the base case and merge(fig17c). This shows that in end-to-end training on classification, the embedding of a token doesn't work like those word2Vec methods which try to assign a token's representation by surrounding tokens and all it needs is a unique representation. Thus, from that point of view, token representation reduces to symbolic representation without any project of human meaning on the numerical value. On the other hand, the inspection of the weight of the merge block(Conv1D) also suggested that the layer does learn since the distribution of weight is changed heavily from the normal distribution that centers around zero to a much more flattened distribution(fig17c). But the question of what it learns still remains in mystery.



(a) Random Merge - seq25 Embedding Weight



(b) Random Merge - seq25 Conv1D_1 Weight



(c) Random Merge - seq25 Conv1D_2 Weight

This thesis gives preliminary insight into the existence of structure inside a language model by the token merge and that structure is representing the I language or mental state in the aspect of the theory of mind. But still, there are lots of questions that arise from the experiment, such as the behavior difference of merging with NOUN on 50 and 25 sequence length. In addition, the single data set environment and small model limit the exploration and the result may variate depending on the number of concepts to infer(just 2 classes in this experiment). Furthermore, the effect on merge and the different behavior on random and fixed pattern merge is questionable on why the random case is still able to learn but not for the fixed one. The experiment can be improved by testing more phrases that are not limited to grammar, but also those mental state verbs that are used in psychology studies. In addition, the truth behind the structure is still unclear, as it can be in a syntactic form(tree base), computational state(automata), a combination of the two, or other types of structures(such as causality). Last but not least, the interaction between natural language understanding and generation is still a missing puzzle in the study of the language model and it is hoped that this research on how internal language may influence by external language initial a tiny step in this direction.

References

[BCE$^+$] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4.

[BLGB20] Cindy Beaudoin, Élizabel Leblanc, Charlotte Gagner, and Miriam H. Beauchamp. Systematic review and inventory of theory of mind measures for young children. *Frontiers in Psychology*, 10, jan 2020.

[BM13] Lindsey J. Byom and Bilge Mutlu. Theory of mind: mechanisms, methods, and new directions. *Frontiers in Human Neuroscience*, 7, 2013.

[Cho05] Noam Chomsky. Three factors in language design. *Linguistic Inquiry*, 36(1):1–22, 2005.

[Dąb15] Ewa Dąbrowska. What exactly is universal grammar, and has anyone seen it? *Frontiers in Psychology*, 6, jun 2015.

[DSV15] Ms. Nithya Dr. S. Vijayarani, Ms. J. Ilamathi. Preprocessing techniques for text mining - an overview. *International Journal of Computer Science & Communication Networks,*, Vol 5(1):7–16, Feb 2015.

[Ebe20] Susanne Ebert. Theory of mind, language, and reading: Developmental relations from early childhood to early adolescence. *Journal of Experimental Child Psychology*, 191:104739, mar 2020.

[GG20] Benyamin Ghojogh and Ali Ghodsi. Attention mechanism, transformers, BERT, and GPT: Tutorial and survey. dec 2020.

[GO22] Ángel J. Gallego and Román Orús. Language design as information renormalization. *SN Computer Science*, 3(2), jan 2022.

[HCF02] Marc D. Hauser, Noam Chomsky, and W. Tecumseh Fitch. The faculty of language: What is it, who has it, and how did it evolve? *Science*, 298(5598):1569–1579, nov 2002.

[JWA05] Jodie A. Baird Janet Wilde Astington, editor. *Why Language Matters for Theory of Mind.* Oxford University Press, Inc., 2005.

[Kos] Michal Kosinski. Theory of mind may have spontaneously emerged in large language models.

[LT03] Heidemarie Lohmann and Michael Tomasello. The role of language in the development of false belief understanding: A training study. *Child Development*, 74(4):1130–1144, jul 2003.

[Mer] Stephen Merity. Single headed attention rnn: Stop thinking with your head.

[MN06] Dániel Pap Krisztina Szécsényi Gabriella Tóth Veronika Vincze Mark Newson, Marianna Hordós. *Language, Grammar and Linguistic Theory*, chapter 1.1, page 3. Bölcsész Konzorcium HEFOP Iroda, 2006.

[NOMC11] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, sep 2011.

[PS09] Jennie E. Pyers and Ann Senghas. Language promotes false-belief understanding. *Psychological Science*, 20(7):805–812, jul 2009.

[Sie95] Hava T. Siegelmann. Computation beyond the turing limit. *Science*, 268(5210):545–548, apr 1995.

[TNO$^+$19] Kyohei Tanaka, Isso Nakamura, Shinri Ohta, Naoki Fukui, Mihoko Zushi, Hiroki Narita, and Kuniyoshi L. Sakai. Merge-generability as the key concept of human language: Evidence from neuroscience. *Frontiers in Psychology*, 10, nov 2019.

[Wim83] H Wimmer. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1):103–128, jan 1983.

Appendix

I. Glossary

	Test Set Accuracy % (NOUN,VERB,ADJ)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
5	0.66	0.65	0.679	0.692	0.042
10	0.66	0.669	0.68	0.69	0.021
25	0.681	0.711	0.725	0.730	0.019
50	0.727	0.729	0.761	0.766	0.037

Table 15. Test Accuracy Details on Always Wrong with only 1 POS Tag.

	Test Set Accuracy % (ADJ+NOUN)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
5	0.642	0.637	0.657	0.669	0.032
10	0.654	0.651	0.675	0.685	0.034
25	0.699	0.701	0.719	0.731	0.03
50	0.722	0.717	0.765	0.771	0.053

Table 16. Test Accuracy Details on Always Wrong with ADJ NOUN .

	Test Set Accuracy % (Others)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
5	0.66	0.655	0.676	0.687	0.032
10	0.654	0.651	0.675	0.685	0.034
25	0.699	0.702	0.722	0.731	0.029
50	0.744	0.754	0.768	0.771	0.017

Table 17. Test Accuracy Details on Always Wrong (Others) .

	Test Set Accuracy % (Random)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
5	0.589	0.548	0.605	0.642	0.094
10	0.599	0.572	0.606	0.637	0.065
25	0.616	0.594	0.637	0.655	0.061
50	0.673	0.669	0.694	0.705	0.036

Table 18. Test Accuracy Details on Random Merge .

	Test Set Accuracy % (VERB)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
5	0.595	0.566	0.612	0.630	0.064
10	0.585	0.550	0.604	0.643	0.093
25	0.574	0.535	0.571	0.600	0.065
50	0.436	0.438	0.489	0.490	0.052

Table 19. Test Accuracy Details on Merge with VERB .

	Test Set Accuracy % (ADJ)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
5	0.597	0.575	0.608	0.634	0.059
10	0.557	0.528	0.568	0.604	0.076
25	0.489	0.502	0.518	0.525	0.023
50	0.418	0.390	0.488	0.491	0.101

Table 20. Test Accuracy Details on Merge with ADJ.

	Test Set Accuracy % (NOUN)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
10	0.532	0.499	0.530	0.581	0.082
25	0.422	0.407	0.489	0.489	0.080
50	0.703	0.704	0.721	0.728	0.024

Table 21. Test Accuracy Details on Merge with NOUN.

	Test Set Accuracy % (PROPN)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
10	0.548	0.512	0.550	0.584	0.072
25	0.586	0.556	0.603	0.620	0.064
50	0.546	0.526	0.543	0.571	0.045

Table 22. Test Accuracy Details on Merge with PROPN.

Top 10 Merge Pattern	
≤ 10	≥ 25
NOUN_NOUN	ADJ_ADJ_ADJ
ADJ_NOUN	ADJ_ADJ
NOUN_VERB	NOUN_NOUN
VERB_NOUN	ADJ_NOUN
NOUN_NOUN_NOUN	NOUN_VERB
ADJ_ADJ	VERB_NOUN
NOUN_ADJ	NOUN_NOUN_NOUN
ADJ_ADJ_ADJ	ADJ_NOUN_NOUN
ADJ_NOUN_NOUN	NOUN_VERB_NOUN
NOUN_VERB_NOUN	NOUN_ADJ

Table 23. Top 10 Random Merge Pattern differences

Wilcoxon signed-rank test			
Tags	Statistic	p-value	Reject Null
Random VS VERB	3655	7.69e-34	Yes
Random VS PROPN	2819.5	3.02e-37	Yes
Random VS ADJ	462	3.02e-47	Yes
VERB VS PROPN	20583	0.718	No
ADJ VS VERB	11327.5	2.05e-11	Yes
ADJ VS PROPN	10129.5	2.72e-14	Yes

Table 24. Signed-rank test on seq len 25

	Test Set Accuracy % (ADJ+NOUN)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
10	0.55	0.509	0.541	0.593	0.084
25	0.389	0.362	0.490	0.490	0.128

Table 25. Test Accuracy Details on Merge ADJ+NOUN.

	Test Set Accuracy % (VERB+NOUN)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
10	0.461	0.460	0.490	0.490	0.030
25	0.370	0.272	0.445	0.490	0.218

Table 26. Test Accuracy Details on Merge VERB+NOUN.

	Test Set Accuracy % (NOUN+VERB+ADJ)				
seq len	mean	quartile 25%	median	quartile 75%	IQR
10	0.519	0.501	0.514	0.529	0.028
25	0.458	0.489	0.493	0.499	0.01

Table 27. Test Accuracy Details on Merge NOUN+VERB+ADJ.

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Chan Wai Tik**,
(author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Emergent Theory of Mind (ToM) from Token Merging,
   (title of thesis)

   supervised by Kallol Roy.
   (supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Chan Wai Tik
09/05/2023