Detecting and translating language ambiguity with multilingual LLMs

Behrang Mehrparvar
Institute for Logic, Language and Computation
University of Amsterdam
Amsterdam, the Netherlands
behrang.mehrparvar@student.uva.nl

Sandro Pezzelle
Institute for Logic, Language and Computation
University of Amsterdam
Amsterdam, the Netherlands
s.pezzelle@uva.nl

Abstract

Language is one of the most important landmarks in humans in history. However, most languages could be ambiguous, which means the same conveyed text or speech, results in different actions by different readers or listeners. In this project we propose a method to detect the ambiguity of a sentence using translation by multilingual LLMs. In this context, we hypothesize that a good machine translator should preserve the ambiguity of sentences in all target languages. Therefore, we investigate whether ambiguity is encoded in the hidden representation of a translation model or, instead, if only a single meaning is encoded. The potential applications of the proposed approach span i) detecting ambiguous sentences, ii) fine-tuning existing multilingual LLMs to preserve ambiguous information, and iii) developing AI systems that can generate ambiguity-free languages when needed.

1 Introduction

Language ambiguity according to (Ceccato et al., 2004) is defined as the potential of different actions as response to a single text by different people, based to their interpretations (see figure 1 for a conceptual illustration).

Several research studies have been focusing on the ambiguity of language. (Wang, 2011) have studied lexical and syntactic ambiguity in Korean language. They propose adding new words as a solution for these types of ambiguities. (Ceccato et al., 2004) have proposed a prototype for an ambiguity identification tool. Furthermore, (Yadav et al., 2021a) have proposed a comprehensive taxonomy of different types of language ambiguities.

As of the well-known example of "Jane saw the man with the telescope.", in many languages including English, sentences do not always correspond to a unique set of possible behaviors and

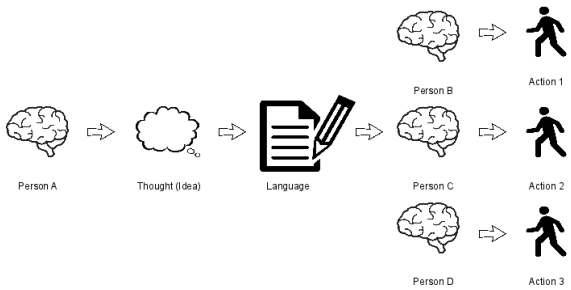


Figure 1: Conceptual illustration of language ambiguity defined as different actions as response of a single text by different people, according to their interpretations.

actions by the readers/listeners which implies language ambiguity. Table 1 provides a list of different types of language ambiguities including examples and the unambiguous version of them.

The goal of this project is investigating and exploring whether it could be possible to provide an LLM translation-based approach to detect language ambiguity in text.

Language ambiguity brings up misunderstandings and conflicts in real world interactions such as political, commercial and cultural interactions. This misunderstanding can lead to either waste of huge amount of time in negotiation between the parties for conflict resolution or even in worse case results in conflicting actions.

Detecting ambiguous statements in global interactions can be a critical problem that requires scientific research in providing solutions. In this project, we provide an LLM translation-based solution for detecting ambiguous statements in language. One major organization that can benefit from the proposed research is the United Nations (UN) where different countries with different languages interact with each other. Considering automatic translation in such organizations where a speech/text is supposed to be translated

Type of ambiguity	Example	Unambiguous version
Lexical	“Give me the bat!”	“Give me the baton!”
Syntactic	“The professor said on Monday he would give an exam"	“The professor said that on coming Monday he would give an exam"
Semantic	“Jane saw the man with a telescope”	“Jane saw the man by using a telescope”
Pragmatic (scope)	“I like you too!”	“I like you too like others do!”
Vagueness (generality)	“The prof said she would give us all A’s.”	“The prof said the TA would give us all A’s.”
Conceptual translational	“Proposal” to “voorstel” and “aanzoek”	“Research proposal”

Table 1: Examples of different types of language ambiguities (Yadav et al., 2021b) and the unambiguous version.

to many languages, detecting and informing the potential ambiguities to both speaker/writer an the listener/reader, would prevent potential misunderstandings, tedious negotiations and conflicting actions between the nations and parties in long term.

The main research questions being investigated in this project are:

Question 1: *Can a state-of-the-art Transformer-based MT model properly encode whether a sentence in the source language is (non-)ambiguous?*

Question 2: *Are both semantic validity and ambiguity preserved by the translation when the sentence is translated into a target language, and then translated back?*

Question 3: *Can we predict the ambiguity of a sentence by translating it into another language looking at the learned hidden representations?*

2 Related work

Before explaining the proposed approach, we provide a brief overview of the related literature, consisting of ambiguity in NLP, ambiguity in machine translation and an overview of multilingual LLMs.

2.1 Ambiguity and underspecification in NLP/LLMs

Language ambiguity, as a subset of semantic underspecification (Pezzelle, 2023), is introduced as the possibility for a linguistic signal to convey only part of the information needed for communication to succeed. The author has investigated how multi-modal models deal with semantic underspecification and how communicative approaches would provide solutions to this type of tasks. In (Hutchinson et al., 2022) the authors also investigated semantic underspecification between text and image. They studied a taxonomy of the family of multi-modal tasks and provided a list of risks and concerns regarding ambiguity in multi-modal text and image tasks.

(Liu et al., 2023) have proposed a benchmark for evaluating pretrained language models to recognize ambiguity and disentangle possible meanings. They capture the ambiguity of the sentences through their entailment relations with other sentences. They have covered different types of ambiguities including pragmatic, lexical, syntactic, scopal, coreference, figurative and other ambiguities.

Furthermore, in (Wildenburg et al., 2024), the authors use perplexity measure to identify Underspecified sentences from the pairs in the DUST dataset. Based on (Egg, 2010), they define four types of underrepresented sentences. They perform experiments on five pre-trained language models and compare the proportional perplexity for pair of sentences in the DUST dataset for each

type of underspecification. They label the underspecified and more specified sentences in a pair with a predefined prompt and then produce another version with switched labels. The intuition behind their proposed solution is that the version with correct labels would have higher probability and therefore lower perplexity.

Although similar to our approach, these papers also recognize ambiguity in sentences, however, our method does not rely on the context, entailment or unambiguous pair of the sentences.

2.2 Ambiguity in machine translation

Language ambiguity is a major factor studied in machine translation. The difficulty not only relies in the nature of the ambiguous sentence by itself, but also the translation mechanism might lack the ability to transfer the ambiguity in semantics for instance, to the target sentence. Regarding language ambiguity in machine translation, several studies have been conducted, some of which are mentioned in this section.

In (Baker et al., 1994) the authors propose a source language analyzer component in their machine translation system that incorporates a controlled lexicon, a controlled grammar and a semantic domain model. Each of these components can be either disambiguated by the author, domain corpus or general corpus. Based on that, they define six test disambiguation methods and compare their disambiguation results in translation. Unlike their approach, our method is not rule-based and hard-coded which results in a more flexible ambiguity detection method. However, in this project, we do not provide direct solutions for disambiguation.

In (Alzeebaree, 2020) they categorize machine translation methods into machine assisted human translation (MAHT), human assisted machine translation (HAMT) and fully automatic machine translation (FAMT). According to this paper, machine translation strategies can also be categorized into direct, transfer and interlingua approaches. The direct method could be either composed of morphological analysis of the source language input text, performing bilingual dictionary lookup or local reordering of the target language. In the transfer approach, for each pair of languages, there is a separate transfer mechanism should be defined, while in the interlingua approach, it is assumed that the source language can be converted to a representation, common to more than one language. As of future work, our method can be considered as a HAMT solution in which the user is asked to disambiguate detected ambiguous sentences in the input text. Also, the machine translation model we use is trained based on interlingua approach.

One of the key points in dealing with ambiguity in translation is choosing the representation of the ambiguous sentence. (Emele and Dorna, 1998) suggest using packed F-structure representations (a form of hierarchical recursive representation) to preserve the ambiguities between source and target language. In cases where the target language is not capable of preserving the ambiguity, the authors propose local disambiguation conducted by human. In (Boguslavsky et al., 2005) the authors propose a rule-based machine translation system that first builds a morphological structure and dependency tree structure. Based on certain predefined rules, the structures are translated to the target language structures and ambiguities are detected accordingly. Afterwards, by interactive disambiguation, the user is able to select the correct intention from the source language. Our approach however represents the sentences in forms of vector representations in the LLM but still do not directly rely on these representations in detecting ambiguity.

In (Sammer et al., 2006) the authors propose using human assistance in lexical ambiguity resolution in machine translation. They develop a system composed of a controlled language lexicon composed of words, word senses, their translations and a short, intuitive gloss or set of clue words to help the user select the correct word sense during interaction with the machine translation system. Contrary to that, we do not require a predefined lexicon for detecting ambiguous words. (Měchura, 2022) investigates gender, number and formality ambiguities in translation. In these cases, according to the paper, the machine translator either decided a random or statistically biased translation. They suggest that in such cases a tool need to be built that first, recognizes unsolved ambiguities and second, asks the human the right questions to disambiguate the text manually. As mentioned before, in future work, we also follow this approach in which asks the human to disambiguate the detected ambiguities in the sentence during machine translation.

2.3 Multilingual large language models

With the advent of Transformer-based language models multilingual models have been proposed. These models which are trained with data from many languages can perform machine translation among many other NLP tasks with higher performance, compared to traditional approaches.

As mutilingual LLMs are trained on multiple languages, the mechanism of how these models perform certain tasks have been recently studied. Knowing the internal mechanism could provide us insight about the ambiguity encoded in the representation of the hidden layers of the LLM.

(Choenni et al., 2023) have studied how individual languages in multilingual LLMs benefit from each other as in cross-lingual sharing in the data level. Furthermore, in (Zhang et al., 2023b) the authors studied how knowledge transfer happens in multilingual LLMs during translation while limited multilingual training data leads to advanced multilingual capabilities. (Liu et al., 2024) have studied the connections of multilingual activation patterns in LLMs at the level of language families. Similar to (Tang et al., 2024), they have discovered (non-)language-specific neurons in the LLMs.

In (Xu et al., 2024) the authors explore human value concepts and study them in terms of cross-lingual inconsistency, distorted linguistic relationships, and unidirectional cross-lingual transfer between high- and low-resource languages. Finally, (Zhao et al., 2024) have studied multilingual LLMs across the layers of the model and realized that the first layers understand the questions by converting the multilingual input to English, the intermediate layers perform problem-solving, mainly in English, and in the last layers, the models generate the response according to the original language.

Finally, multilingual LLMs have been used in different NLP tasks and applications. Although machine translation is the main NLP task that we focus on, we provide a brief overview of other recent developments of using LLMs for other NLP tasks for reference.

In (Mendonça et al., 2023) the authors augmented existing English dialogue data using machine translation to improve a multilingual pretrained LLM for dialogue evaluation. Similarly (Mittal et al., 2023) built a novel dataset for claim span identification. In (Qi et al., 2023) the authors study cross-lingual consistency of factual knowledge and propose a metric to evaluate knowledge consistency across languages independently from accuracy. (Tanwar et al., 2023) study cross-lingual in-context learning.

In (Wang et al., 2023) the authors perform model-agnostic knowledge editing in multilingual settings where the new knowledge is supplied in one language, but the query is in a different language. (Zhang et al., 2023a) performed a benchmarking of different tasks including sentiment analysis, machine translation, summarization and wordlevel language identification on multilingual LLMs. Finally, (Zhu et al., 2023), (Zhu et al., 2024) and (Gao et al., 2024) have studied multilingual machine translation in LLMs.

3 Proposed method

The goal of this project is to provide an approach in detecting language ambiguities using multilingual large language models (LLMs). As mentioned before, (Ceccato et al., 2004) define language ambiguity by the different potential actions that could be performed by different agents. Inspired by this idea, we propose language translation as an action performed by LLM agents. Accordingly, we propose a four step approach in detecting language ambiguity illustrated in figure 2:

Step 1: Translate the input text from the source language into the target languages using the LLM. Then extract the hidden representation of the state of the LLM as a vectors.

Step 2: Translate the output texts of the first step from the target language into the source language using the same LLM. Then extract the hidden representation of the state of the LLM as a vector.

Step 3: Compute a function that maps the two representations above.

Step 4: Compute the overall measure of ambiguity based on the properties of the mapping function.

As shown in figure 2, considering n different meanings for input text t_A and m different interpretation of the output text t_B, in worst case we would have $n \times m$ different translation meaning pairs which complicates the problem of ambiguity in translation.

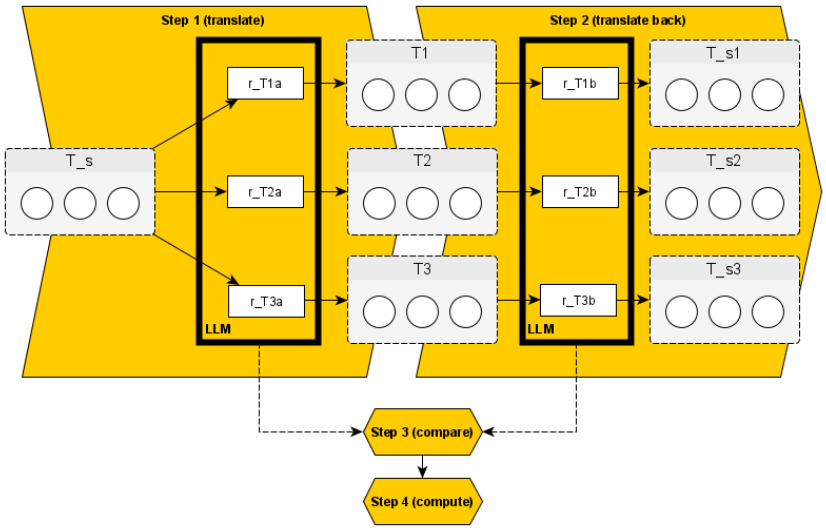


Figure 2: Proposed approach in language ambiguity detection using LLM translation consisting of four steps: 1) translating the text into the target languages, 2) translating back the new texts into the source language, 3) comparing the pairwise representations, 4) computing the overall measure. The boxes represent texts with different ambiguous meanings shown with circles in them and the rectangles show the hidden layer representations of the LLM translation model.

The LLM works as a function $f(.)$ defined in equation (1):

$$r \mapsto f(t, l_s, l_t) \tag{1}$$

where t is the input text, l_s is the source language, l_t is the target language and r is the vector representation of the hidden state of the LLM. By applying the translation function $f(.)$ in steps 1 and 2 listed above, the representation vectors can be found as in equation (2):

$$\begin{aligned} r_A &= f(t_A, l_1, l_2) \\ r_B &= f(t_B, l_2, l_1) \end{aligned} \tag{2}$$

where t_A is the input text and t_B is the generated output text from the translation using the LLM in step 1.

Despite the complicated architecture of current LLMs, all these models are based on the idea of artificial neurons mathematically defined as in equation (3):

$$y = \sigma(w_1 x_1 + w_2 x_2 + ... + w_n x_n + b) \tag{3}$$

where w_i's represent the elements of the weight vector W, b is the bias, x_i's are the elements of the input vector X and y is the output.

As shown in equation (3), the output value y is a weighted combination of all elements of the input vector. Moving to a more complex architecture of LLM, considering function $f(.)$ in equation (1), similarly the output representation r is also a complex combination of all three input vectors t, l_1 and l_2. Therefore, it is not easy to directly extract the effect of text t on r.

The hidden representation r consists of a distributed representation of multiple factors, not only including the semantics. Therefore, as it is not possible to directly and manually extract the representation of the input text t from the representation r, and also because of the continuous space of the representation r, we can not directly compare the two representations r_A and r_B to detect ambiguity in text. Therefore we propose a different approach in detecting ambiguity.

In first step we define a function $g(.)$ that maps the two representations to each other as illustrated in equation (4):

$$r_B = g(r_A) \tag{4}$$

where r_A and r_B are the representations found from equation (2) and $g(.)$ is the mapping function.

In order to find the function $g(.)$, we learn a simple auto-encoder with a single hidden layer of size s_H, input size of s_A and output size of s_B. Note

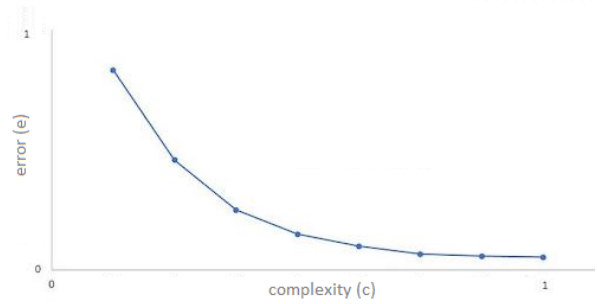


Figure 3: Elbow plot of mapping error e over network complexity c for the network implementing the mapping function $g(.)$

that as the translation in steps 1 and 2 are both performed using the same LLM, we have:

$$s_A = s_B \tag{5}$$

We define the function $c(.)$ as complexity of the function $g(.)$ as follows:

$$c(g) = s_H / s_A \tag{6}$$

where as explained before, s_H and s_A are the sizes (number of neurons) in the hidden layer H and input r_A of the neural network implementing the $g(.)$ function.

The error of the network implementing $g(.)$ is defined as normalized mean squared error (NMSE) of the elements of the two representations r_A and r_B defined in equation (7):

$$\begin{aligned} e(r_A, r_B) &= \frac{1}{s_A} \sum_{i=0}^{s_A} \frac{(r_A^i - r_B^i)^2}{\overline{r_A}\,\overline{r_B}} \\ \overline{r_A} &= \frac{1}{s_A} \sum_{i=0}^{s_A} r_A^i \\ \overline{r_B} &= \frac{1}{s_B} \sum_{i=0}^{s_B} r_B^i \end{aligned} \tag{7}$$

where r_X^i is the i'th element of representation r_X and s_X is the size (number of neurons) of r_X.

Having chosen the right learning rate and number of training epochs for the network implementing $g(.)$, for each text t_A in the input dataset, we can evaluate the translation error $e(.)$ for each setting of the network complexity $c(.)$ and plot them as illustrated in figure 3.

The main idea for using an auto-encoder is based on the assumption that: *An Auto-encoder should behave differently in mapping ambiguous and unambiguous sentences.* In other words, we want to investigate how reconstruction error changes over model size and target language is predictive of sentence ambiguity which will be explained in more details as follows.

In this project, we plan to experiment with two approaches to measure the ambiguity based on the mapping function; the engineering approach and the machine learning approach.

3.1 Engineering approach

In the engineering approach, we propose a formula for computing the ambiguity of the sentence based on the mapping function.

As shown in figure 3, having considered the trade-off between complexity and error, we assume that the resulting plot would look like an elbow shape. In an elbow shaped curve, as defined in (Salvador and Chan, 2004), a knee point is a point with a maximum curvature. This point intuitively reflects the optimum point for some decision and a trade-off between the (reverse) benefit (vertical y axis) and the cost (horizontal x axis). In our case, by increasing the cost of complexity, we achieve less error and vice versa.

According to (Satopaa et al., 2011), the curvature of a function $f(.)$ is defined in equation (8):

$$K_f(x) = \frac{f''(x)}{(1 + f'(x)^2)^{1.5}} \tag{8}$$

Based on that, in (Satopaa et al., 2011) the authors have proposed the Kneedle algorithm to find the maximum curvature of a curved function from a dataset representing a distribution of data points.

We define "translation ambiguity" of text t_A as follows:

$$\begin{aligned} m &= \sqrt{(1 - \dot{x})(1 - \dot{y})} \\ \dot{x} &= \arg\max_x K_f(x) \\ \dot{y} &= c(\dot{x}) \end{aligned} \tag{9}$$

The use of geometric mean of x and y instead of arithmetic mean is explained in (Nunes et al., 2016) where the authors used this measure for evaluating the sustainability of socio-ecological systems as a trade-off between input dimensions of the elbow function which is conceptually similar approach to our problem.

After computing the translation ambiguity m for each language l, we compute the overall ambiguity α as follows:

$$\begin{aligned} \alpha_1 &= 1 - \frac{2}{n}\sum_{i\in L}|m_i - 0.5| \\ \alpha_2 &= \frac{1}{n^2}(\sum_{i\in L}\sum_{j\in L}|m_i - m_j|) \\ \alpha &= 0.5\alpha_1 + 0.5\alpha_2 \end{aligned} \tag{10}$$

where m_l is the ambiguity in language l and n is the number of target languages in L.

The first terms in the α equation indicates the degree of ambiguity/unambiguity of each translation and the second term defines the distance between each pair of translation ambiguities to different target languages. The output of α is low when either all translated texts are ambiguous or unambiguous together.

3.2 Machine learning approach

In the machine learning approach, we propose using a simple neural network model to predict ambiguity using the data points in the elbow chart if figure 3 as input in a supervised manner.

3.3 Experiments

Having a mixed collection of ambiguous and unambiguous English sentences from the Dataset of semantically Underspecified Sentences by Type (DUST)[1], as in equation (11), we use a multilanguage translation model such as Facebook M2M100[2] (Fan et al., 2020) to translate each sentence from English to other possible languages and translate them back to English. We use German, Greek, Persian, Spanish, French, Hindi, Italian, Korean, Dutch, Russian, Turkish, Croatian, Romanian and Chinese as our target languages. After translation, we extract the hidden states of the LLM for the two translation steps as defined in equation (11):

$$\begin{aligned} T_A &= \{t_A^j\} \\ R_A &= \{r_A^j\} \\ R_B &= \{r_B^j\} \end{aligned} \tag{11}$$

After learning the network for function $g(.)$, we feed all the r_A's to the network and capture the outputs r'_B's. Using equations (6) and (7), we find the complexity and error for each sample and each network size. Figure 4 shows the mapping between function for an ambiguous sentence along with its unambiguous version.

After finding the mapping functions, using equation (10) or using a classification method, we find the ambiguity of each sentence $t_A^i \in T_A$. For classification we used a neural network or logistic regression. Further detail about the classification experiments are explained in section 4.2.

As an analysis of the experiment before, for the misclassified samples, we ask native annotators of two languages out of the set reported above to verify if the corresponding sentence in the target language is (A) semantically valid and (B) (non-)ambiguous. Semantic validity is verified by asking the human user whether the sentence is correctly translated, and ambiguity is verified by asking whether the translated sentence is (still) ambiguous or not.

3.4 Evaluation

In the proposed project, as mentioned before, we translate ambiguous ad unambiguous English sentences to several other languages and investigate whether the meaning has change through analysis of the hidden states of the multilingual LLMs.

According to our experiment, if we are able to get high accuracy in predicting ambiguity of ambiguous sentences, we could conclude that the MT model was able to properly encode the ambiguity in its hidden representations (research question 1). Furthermore, the accuracy answers the research question 3 as well.

Based on the human analysis, we would be able to identify the source of misclassification of ambiguity, whether the MT model or the target language itself (research question 2).

4 Results

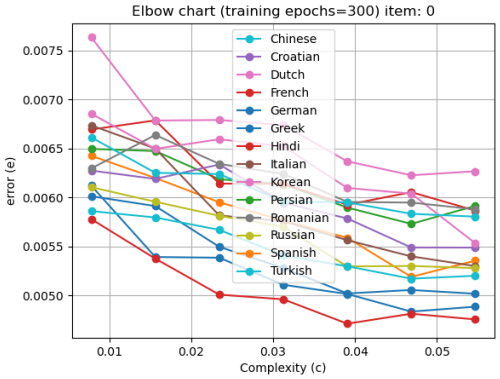
In this section, we provide the results of our experiments.

4.1 Discriminability

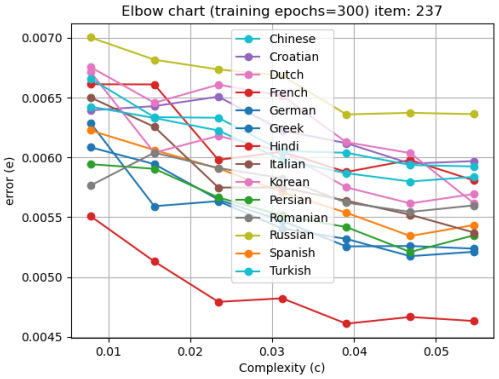
In the first step of our analysis, we examined the discriminability of reconstruction error of the best auto-encoder per each language for ambiguity. Figure 5 illustrates the distribution of reconstruction error along languages for each class. In order to evaluate the discriminability, we performed

[1] https://github.com/frank-wildenburg/DUST

[2] https://huggingface.co/facebook/m2m100_418M



(a) "Andrei picked up the chair or the bag and the telescope" (ambiguous)



(b) "Andrei picked up the chair, or both the bag and the telescope" (unambiguous)

Figure 4: Illustration of the mapping function for an ambiguous sentence along with its unambiguous version

Language	t-test	p-value
German	-0.341	0.33
Greek	0.510	0.610
Persian	-1.95	0.051
Spanish	-0.087	0.931
French	-1.072	0.285
Hindi	1.828	0.069
Italian	-0.821	0.413
Korean	1.864	0.063
Dutch	-2.253	**0.025**
Russian	-0.905	0.366
Turkish	-1.557	0.121
Croatian	-1.034	0.452
Romanian	-1.594	0.112
Chinese	-3.307	**0.001**

Table 2: T-test statistics indicating discriminability of reconstruction error of best auto-encoder for ambiguity. We consider $pvalue < 0.05$ as statistically significant.

t-test statistics. The detailed results are listed in table 2.

Based on the results of t-test, we can conclude that mean reconstruction error for separate target languages are not informative enough to discriminate ambiguous and unambiguous sentences, except for a limited number of languages.

4.2 Classification

In order to determine the most informative variables for classification, we performed several experiments, each including a different setting composed of the options listed in table 3.

Setting	Options
Input type	- Equation (10) - Single language across all auto-encoder models - All languages only for the best auto-encoder - Only relations across languages - Whole mapping functions
Input variable	- α in eq. (10) - Reconstruction error - Reconstruction error difference
Output	- Ambiguous vs Unambiguous - Ambiguity type
Model	- Logistic regression - Neural network
Cross-validation	- 10-fold

Table 3: Experiment settings for ambiguity classification

Table 4 shows the results of classification in all experiment settings. The detailed analysis of the findings for these experiments are provided in section 5.

4.3 Source of Ambiguity

After classifying the data, we investigated the source of misclassificaion using annotation for Italian and Persian language. Accordingly, we found machine translation and incapability of the

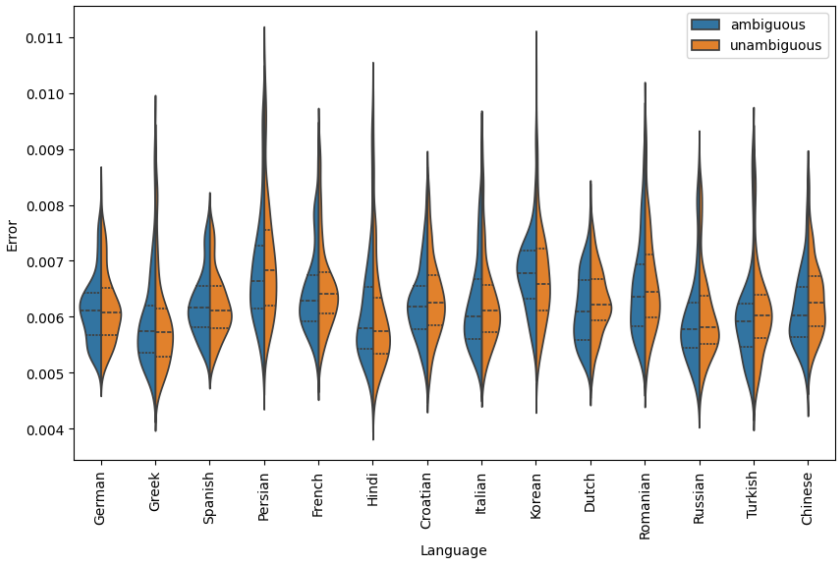


Figure 5: Discriminability of reconstruction error along language for the best auto-encoder. Languages other than Dutch and Chinese are not significantly separable according to table 2.

Input	Input variable	Output	Model	Accuracy	F-Measure
Equation (10)	α	Amb. Vs unamb.	LR	52.53%	0.525
Persian	Differences	Amb. Vs unamb.	LR	57.81%	0.578
Best AE	Values	Amb. Vs unamb.	LR	66.67%	0.667
Along languages	Differences	Amb. Vs unamb.	LR	85.87%	0.859
Whole	Differences	Amb. Type	LR	92.83%	0.928
Whole	Differences	Amb. Vs unamb.	LR	88.19%	0.882
Best AE	Values	Amb. Vs unamb.	NN	73.21%	0.732
Whole	Values	Amb. Vs unamb.	NN	81.99%	0.820
Whole	Values	Amb. Type	NN	78.26%	-
Whole	Differences	Amb. Type	NN	93.04%	0.925
Whole	Differences	Amb. Vs unamb.	NN	94.94%	0.949

Table 4: Classification results for different settings. For classifying ambiguous vs unambiguous sentences the chance level accuracy is 50.0% and for ambiguity type it is 36.58%

target language as the sources of misclassification. Figure 6 illustrates these results.

From the misclassified sentences, considering two target languages (Italian and Persian) we found the following outcomes:

- Ambiguity was lost in 44.68% of the Italian and 51.02% of the Persian target sentences (out of misclassified ambiguous sentences).
- From the misclassified sentences that the ambiguity was lost, in Italian target language, 85.71% of the loss was because of the translation model and the sentence could be written in an ambiguous sense by a native human. However, none of the loss of ambiguity was because of the translation in Persian target language and the native Persian human was also unable to translate the ambiguity into the target language due to the innate difference between English and Persian languages.
- From the unambiguous misclassified sentences, in 7.69% of the cases, ambiguity was introduced in Italian translation, none of which was because of wrong translation by the machine, but because of the innate difference between the target language and English. This percentage increases in Persian to 26.67% of the unambiguous misclassified sentences which was similarly due to the innate difference in languages and not because of machine translation.

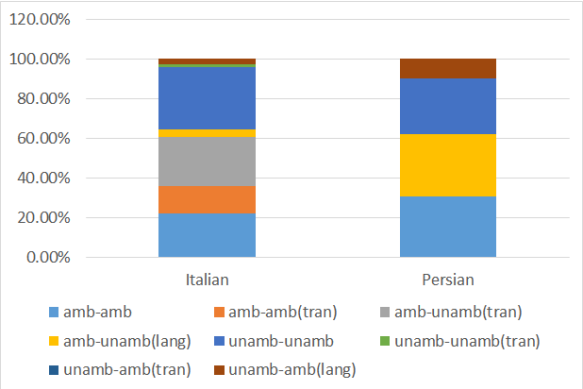


Figure 6: Misclassified samples distribution - format: source-target(problem): *amb*: ambiguous, *unamb*: unambiguous, *tran*: source of misclassification is wrong machine translation, *lang*: source of misclassification is target language incapability in transfering ambiguity sense

- We can conclude that 68.49% of the misclassified sentences in total were correctly translated in terms of ambiguity in Italian while 58.23% in Persian, from which 78.26% (for Italian) and 0.0% (for Persian) was because of machine translation problem.

5 Discussion

Based on the results of our classification experiments shown in table 4, we achieved the following findings:

1. The engineering approach proposed in equation (10) was not able to predict ambiguity (accuracy is 52.53%).

2. Single language translation is not informative enough in predicting ambiguity. By moving from one language (Persian) to all languages we achieved 85.87% accuracy (from 57.81%). This could be due to adding more features to the classification algorithm.

3. Single best auto-encoder is not informative enough in predicting ambiguity. The accuracy have changed from 66.67% to 88.19% by introducing more auto-encoder models even with lower complexities. Adding more features about the gradual change over the complexity of the auto-encoder model could explain this phenomenon.

4. Adding reconstruction error differences between languages improves accuracy. By adding this information we achieved 88.19% accuracy compared to 85.87%. Accordingly, adding more features about the properties of the mapping function mesh actually improved the accuracy.

5. Reconstruction error differences is more informative than their values. This phenomena can be observed from the results by improving from 81.99% to 94.94% accuracy. We can conclude that the shape of the mapping function is informative not the position of it. However, we would expect that a nonlinear complex classifier would also be able to pick this feature.

6. A simple linear model can perform relatively close to a complex neural network model. The accuracy of the complex model was 94.94% compared to 88.19% for the linear model. Learning more complex and nonlinear features actually helped the classification.

7. Predicting more detailed classes improves the accuracy in linear models. For the linear model, the accuracy have changed from 88.19% (F-measure 0.820) to 92.83% (F-measure 0.928) by changing to multi-class classification. It can be explained by classifying more detailed regions in the misclassified regions. For more details on the distribution of te classes along the main two principle components, refer to figure 7. For the neural network however, the classification result decreased from 94.94% to 93.04% by moving to multi-class classification. Compared to the increase of accuracy in the linear model, we can explain that the neural networks have been already able to learn the nonlinear boundaries in the input space and already got a high accuracy in two-class classification.

Moving back to our initial research questions, based on the classification accuracies in table 4, we can claim that it is possible to predict sentence ambiguity using machine translation. However, we can not claim that the semantic validity and ambiguity is preserved by translation for all target languages and it highly depends on the language. Finally, we conclude that the ambiguity of the sentence is actually encoded in the hidden representation of the LLMs, as the ambiguity is predictable

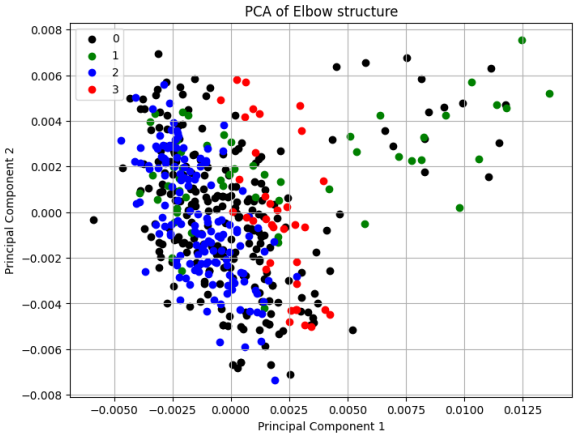


Figure 7: Data distribution over two main principle components

from these representations.

The main contribution of the project is proposing an approach that can predict ambiguity of the sentences, without direct use of semantics. As explained in section 3 this feature is achieved by classifying the ambiguity based on the shape of the mapping function and not directly from the semantics. As a consequence, the algorithm does not require extensive training data to cover the whole semantic space of the source language. Furthermore, the approach is potentially much more generalizable to unseen sentences with unseen semantics. Also, the model would be robust to changes to the input distribution as it is independent of the semantics.

6 Future work

One future direction of the proposed method is to investigate in more details the source of misclassification for all fourteen target languages other than Italian and Persian. Other than that, detecting the source of ambiguity in sentences in terms of words could be an interesting direction. Furthermore, extending the method to different source languages other than English and investigating which step of the proposed approach should be tailored accordingly could also be considered as future work.

One of the potential applications of an ambiguity detection method could be in automatic translation of critical documents e.g. legal, political, commercial, etc. where the user is asked to clarify the ambiguity of the source language manually, to prevent misunderstanding and potential conflicts.

Fine-tuning existing multilingual large language models to preserve ambiguity in sentences could be another potential application of the proposed method..

Finally, the trained classifier model can potentially be used as a partial loss function for designing and optimizing ambiguity-free AI-generated human languages investigated at Synaptosearch[3]. In order to do so, for each input sentence generated by the AI, the ambiguity is measured using the model and the gradient with respect to the input is calculated and used to optimize the loss function term related to ambiguity.

Ambiguity can be considered of a strength of the language in cases such as providing efficient means of communication or when it is used as amphibology in literature. However, in critical political, commercial and cultural cases unintended ambiguity results in misunderstandings and conflicts. The outcome of the misunderstanding could lead to spending a lot of time in negotiation to elaborate the meaning, or in worse case conflicting actions.

In this project, we introduced an LLM translation-based approach in detecting ambiguity in language. We believe that that proposed approach not only provides a detection algorithm, but also opens an avenue for developing an AI-generated global human language with features such as being ambiguity-free.

References

Yaseen Alzeebaree. 2020. Lexical and Structural Ambiguity in Machine Translation: An Analytical Study. *Eastern Journal of Languages, Linguistics and Literatures*, 1(1).

Kathryn Baker, Alexander Franz, Pamela Jordan, Teruko Mitamura, and Eric Nyberg. 1994. Coping with ambiguity in a large-scale machine translation system. In *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*.

Igor M Boguslavsky, Leonid L Iomdin, Alexander V Lazursky, Leonid G Mityushin, Victor G Sizov, Leonid G Kreydlin, and Alexander S Berdichevsky. 2005. Interactive resolution of intrinsic and translational ambiguity in a machine translation system. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 388–399. Springer.

[3]https://synaptosearch.com/

Mariano Ceccato, Nadzeya Kiyavitskaya, Nicola Zeni, Luisa Mich, and Daniel M Berry. 2004. Ambiguity identification and measurement in natural language texts. Publisher: University of Trento.

Rochelle Choenni, Dan Garrette, and Ekaterina Shutova. 2023. How do languages influence each other? Studying cross-lingual data sharing during LLM fine-tuning. *arXiv preprint arXiv:2305.13286*.

Markus Egg. 2010. Semantic underspecification. *Language and Linguistics Compass*, 4(3):166–181. Publisher: Wiley Online Library.

Martin C Emele and Michael Dorna. 1998. Ambiguity preserving machine translation using packed representations. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 365–371.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

Pengzhi Gao, Zhongjun He, Hua Wu, and Haifeng Wang. 2024. Towards boosting many-to-many multilingual machine translation with large language models. *arXiv preprint arXiv:2401.05861*.

Ben Hutchinson, Jason Baldridge, and Vinodkumar Prabhakaran. 2022. Underspecification in scene description-to-depiction tasks. *arXiv preprint arXiv:2210.05815*.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity. *arXiv preprint arXiv:2304.14399*.

Weize Liu, Yinlong Xu, Hongxia Xu, Jintai Chen, Xuming Hu, and Jian Wu. 2024. Unraveling Babel: Exploring Multilingual Activation Patterns within Large Language Models. *arXiv preprint arXiv:2402.16367*.

John Mendonça, Alon Lavie, and Isabel Trancoso. 2023. Towards Multilingual Automatic Dialogue Evaluation. *arXiv preprint arXiv:2308.16795*.

Shubham Mittal, Megha Sundriyal, and Preslav Nakov. 2023. Lost in translation, found in spans: Identifying claims in multilingual social media. *arXiv preprint arXiv:2310.18205*.

Michal Měchura. 2022. A taxonomy of bias-causing ambiguities in machine translation. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 168–173.

Breno Nunes, Roberto C Alamino, Duncan Shaw, and David Bennett. 2016. Modelling sustainability performance to achieve absolute reductions in socio-ecological systems. *Journal of Cleaner Production*, 132:32–44. Publisher: Elsevier.

Sandro Pezzelle. 2023. Dealing with semantic underspecification in multimodal NLP. *arXiv preprint arXiv:2306.05240*.

Jirui Qi, Raquel Fernández, and Arianna Bisazza. 2023. Cross-Lingual Consistency of Factual Knowledge in Multilingual Language Models. *arXiv preprint arXiv:2310.10378*.

Stan Salvador and Philip Chan. 2004. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *16th IEEE international conference on tools with artificial intelligence*, pages 576–584. IEEE.

Marcus Sammer, Kobi Reiter, Stephen Soderland, Katrin Kirchhoff, and Oren Etzioni. 2006. Ambiguity reduction for machine translation: Human-computer collaboration. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 193–202.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. Finding a" kneedle"

in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-Specific Neurons: The Key to Multilingual Capabilities in Large Language Models. *arXiv preprint arXiv:2402.16438*.

Eshaan Tanwar, Manish Borthakur, Subhabrata Dutta, and Tanmoy Chakraborty. 2023. Multilingual llms are better cross-lingual in-context learners with alignment. *arXiv preprint arXiv:2305.05940*.

Weixuan Wang, Barry Haddow, and Alexandra Birch. 2023. Retrieval-augmented Multilingual Knowledge Editing. *arXiv preprint arXiv:2312.13040*.

William Shi-Yuan Wang. 2011. Ambiguity in language. *Korea Journal of Chinese Language and Literature*, 1:3–20.

Frank Wildenburg, Michael Hanna, and Sandro Pezzelle. 2024. Do Pre-Trained Language Models Detect and Understand Semantic Underspecification? Ask the DUST! *arXiv preprint arXiv:2402.12486*.

Shaoyang Xu, Weilong Dong, Zishan Guo, Xinwei Wu, and Deyi Xiong. 2024. Exploring Multilingual Human Value Concepts in Large Language Models: Is Value Alignment Consistent, Transferable and Controllable across Languages? *arXiv preprint arXiv:2402.18120*.

Apurwa Yadav, Aarshil Patel, and Manan Shah. 2021a. A comprehensive review on resolving ambiguities in natural language processing. *AI Open*, 2:85–92. Publisher: Elsevier.

Apurwa Yadav, Aarshil Patel, and Manan Shah. 2021b. A comprehensive review on resolving ambiguities in natural language processing. *AI Open*, 2:85–92.

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, and Alham Fikri Aji. 2023a. Multilingual large language models are not (yet) code-switchers. *arXiv preprint arXiv:2305.14235*.

Xiang Zhang, Senyu Li, Bradley Hauer, Ning Shi, and Grzegorz Kondrak. 2023b. Don't trust ChatGPT when your question is not in English: A study of multilingual abilities and types of LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7915–7927.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do Large Language Models Handle Multilingualism? *arXiv preprint arXiv:2402.18815*.

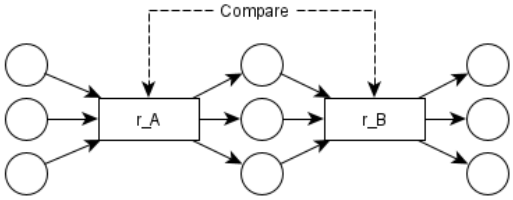
Wenhao Zhu, Shujian Huang, Fei Yuan, Shuaijie She, Jiajun Chen, and Alexandra Birch. 2024. Question Translation Training for Better Multilingual Reasoning. *arXiv preprint arXiv:2401.07817*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.

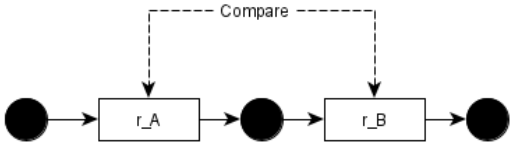
A Possible states in translating ambiguity

Considering several possibilities of translating (un)ambiguous sentences, we summarize 6 states that can be found in table 5 and figure 8.

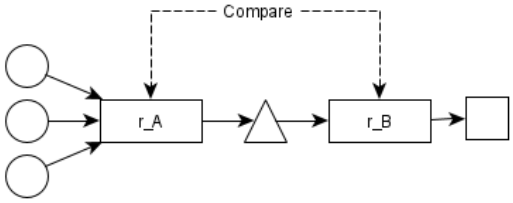
According to figure 8, for unambiguous sentences, state $sU0$ is desirable and for ambiguous source sentences, for all target languages, either of the states $sA0$ or $sA2$ is desirable. In other words, if a sentence is ambiguous, it should be either ambiguous in all target languages, or none of them. This is the intuition behind equation 10 as well.



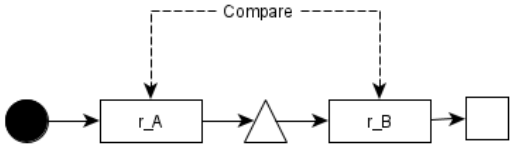
(a) State sA0



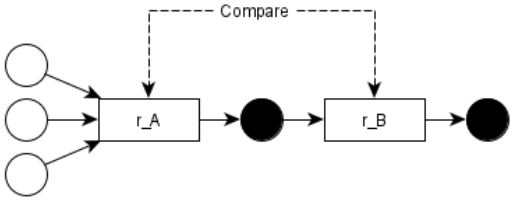
(b) State sU0



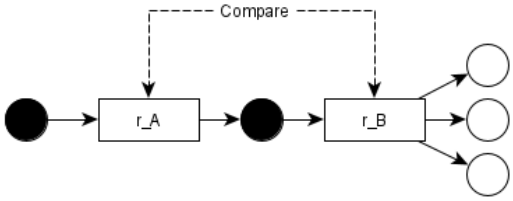
(c) State sA1



(d) State sU1



(e) State sA2



(f) State sU2

Figure 8: Possible states of the 2-step translation approach proposed in the project. White circles indicate certain meanings associated to an ambiguous sentence. Black circles indicate a biased meaning from possible meanings of an ambiguous sentence. Rectangles indicate the internal hidden states of a translation step. Triangles and squares indicate incorrect translations. For detailed description about the possible states refer to table 5.

Table 5: Possible states of the 2-step translation approach proposed in the project.

Tag	Source	Target	Case study	Hyp. Score	Notes
sA0	Ambiguous	Ambiguous	26%	0	Perfect hypothetical translation and rich target language. But score doesn't detect ambiguity.
sA1	Ambiguous	Incorrect	38%	1	Incorrect translation in step 1.
sA2	Ambiguous	Unambiguous	36%	?	If the score is 1, then we could conclude that ambiguity is encoded in the representation and the representation is not biased towards certain meanings. Also reaching this state might be because of the unambiguity in the target language by itself.
sU0	Unambiguous	Unambiguous	30%	0	
sU1	Unambiguous	Incorrect	70%	1	Only one sentence in case study; not reliable statistically.
sU2	Unambiguous	Ambiguous	0%	1	Very rare, but possible.