Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EXPERIMENTAL STUDY OF NEURAL NETWORK-BASED WORD ALIGNMENT SELECTION MODEL TRAINED WITH FOURIER DESCRIPTORS

[1]AMANDYK KARTBAYEV, [2]UALSHER TUKEYEV, [3]SVETLANA SHERYEMETIEVA, [2]ALIYA KALIZHANOVA, [4]BEKTURGAN KALYBEK UULY

[1]Department of Information Systems, Al-Farabi Kazakh National University, Almaty, Kazakhstan

[2]Al Farabi Kazakh National University, Almaty, Kazakhstan

[3]Department of Linguistics, Copenhagen Business School, Copenhagen, Denmark

[4]Institute of Automation and Information Technologies of the National Academy of Sciences of the Kyrgyz Republic

E-mail: [1]a.kartbayev@gmail.com, [2]ualsher.tukeyev@kaznu.kz, [3]lanaconsult@mail.dk, [2]kalizhanova_aliya@mail.ru

ABSTRACT

This paper presents an approach to word alignment selection by Fourier descriptors that were used together with a neural network for image recognition. Word alignment selection is an important problem of statistical machine translation. The recognition of correct word alignment images is a special case of shape recognition. There are various ways of studying image contours experimentally, and we choose the Fourier method of descriptors, which is proved to be effective and easy to implement. The key implementation options and advantages of the method have been considered. From the given information of the contour and the method of its comparison with the references, an algorithm of word alignment selection has been developed. We also set some threshold conditions for more accurate learning of contours and common patterns.

Keywords: *Word Alignment, Machine Translation, Image Recognition, Fourier Descriptors, Neural Networks.*

1. INTRODUCTION

There are numerous studies which examine statistical methods for improving the quality of word alignment. The researchers found that the statistical methods can successfully improve the system score using well-known alignment quality metrics, many of that approaches led to a significant increase in translation performance [1]. The main reason is that the alignment quality metrics well measure the impact of alignment to translation. These well-known metrics are the balanced F-measure and alignment error rate (AER). Earlier, the correlation between these metrics and the BLEU metric was at substantial level [2]. A brief mathematical description of correlation is the coefficient of determination, which is square of the Pearson correlation coefficient. For the alignment experiment conducted on elaborately prepared data set, AER shows a significant correlation with the BLEU score [3].

Current approaches to shape recognition can be described as methods based on the contour, domain, spatial domain and transformation domain. However, general approaches to shape selection are usually divided in methods of processing, one-dimensional functions, shapes approximation, spatial relations features, and shape transformation. Due to their effectiveness and performance, we chose the method of Fourier descriptors from various contour description methods [4].

Since our goal is to find automatically the best word alignments, we would like to confirm the hypothesis that improving the consistency of word alignment can improve machine translation (MT) systems. The IBM alignment model [5] trained on Kazakh-English parallel corpus has been used as a

Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

basis for the experiment. We wish to have better alignments using the image recognition system that effectively selects correct alignments. The given performance metrics are well correlated with human judgments, so we have measured scores of the statistical machine translation system on our new alignments and compared the scores thereof with the baseline scores. As we saw, improved word alignment usually results in a higher BLEU score, in most cases, it is much better than the baseline system.

We want to verify whether our method allows us achieving necessary results in reasonable time and with available computational resources. The dataset of basic 50000 images has been generated and used as training data. We train a neural network to have the same result as statistical models at the end. The experiments presented in this paper help evaluate the correctness of previous models and compare the new method with the traditional one, which has been created with the same initial assumptions.

The rest of the article is organized as follows. Section 2 overviews the word alignment examples, which we use like development data. Section 3 presents Fourier descriptors method with describing in detail all its components, including the individual features of the method. Section 4 describes the experimental work carried out with the neural networks and discusses the results obtained. Section 5 covers the evaluation of the proposed system. Finally, Section 6 concludes and highlights open issues for future research.

2. TRAINING DATA

In this section, we want to determine typical word alignments that are used to train machine translation systems. Word alignment, as produced by the IBM models, is a mapping between source and target words. It has a crucial role in Statistical Machine Translation (SMT) as it is used to learn translation rules in most of the approaches.

We use a tool named Picaro[6] for word alignment visualization. It is a simple visualization tool, which uses a grid style to display alignment information. Although Picaro is a quite simple tool, it gives a quick and clear representation of data and is ideal for automatic grid generation. It is close to the following alignments matrix (Figure 1), which is produced for the English-Kazakh pair of languages.

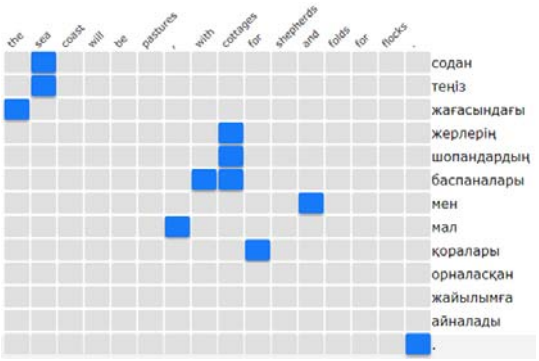

Figure 1: An ordinary Kazakh-English word alignment

Human annotated word alignments often contain M-N alignments, where multiple source words are aligned to multiple target words, and the resulting alignment can be decomposed into even more. The source or target words in word alignment are sometimes not coherent. Unfortunately, existing word alignment models cannot detect these errors, since they do not make assumptions about the alignment structure.

Words alignment defines the minimum number of words in two parallel sentences that correspond to each other, which are called "correspondence". There we present the most typical images of the correct word alignments that are used in the training of the model (Figure 2).

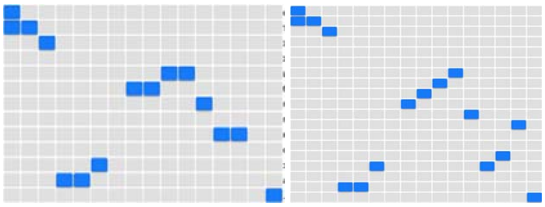

Figure 2: Correct Kazakh-English word alignments

The common pattern of all these word alignments is their tendency to take places along the main diagonal of the alignment matrix. There may be various deviations from diagonal placement, but in an ideal case any alignment sequence can be turned into this type. The accuracy of the visualization shown in both figures above is not critical; otherwise the density of data is relatively critical. The image shows matching words in text boxes between two sentences, using lines to display the connections. Figure 3 shown below is an example of the alignment, where words in parallel sentences are aligned perfectly.

Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

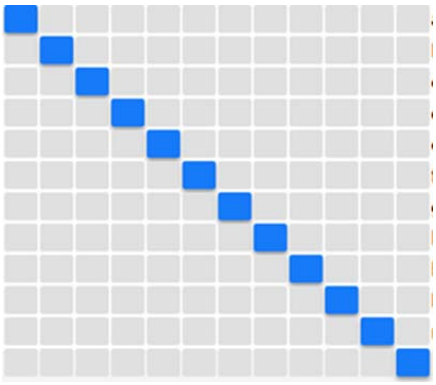

Figure 3: An ideal Kazakh-English word alignment

2.1 Fourier descriptors method

The contour is a closed boundary around the object, which can be represented as an ordered sequence of coordinate pairs:

$$E[e_1, e_2, ..., e_n \mid e = P]; \quad (1)$$

$$P[x, y], \quad (2)$$

where E is an object contour; $e_1, e_2, \ldots, e_n$ – points, forming the contour; n – total number of the points; P – a coordinate pair; x, y – coordinates of the contour points.

The contour is a two-dimensional figure, where each its point is described by a pair of values. The contour of the object can be transformed to a one-dimensional form. It can be more convenient for Fourier transformation and may help to visualize the contour more clearly. To bring the contour to the one-dimensional form, an array of center-of-mass distances is used.

The array of center-mass distances is a one-dimensional discrete ordered sequence, which is equal to the distance from the center of mass of the contour to the point on its boundary. We define this sequence as:

$$R = [r_1, r_2, ..., r_n], \quad (3)$$

where R is an array of center-mass distances; r is a distance from the center of mass to the contour boundary point; n is the total number of points that make up the contour.

The distance from the center of mass to the point on boundary of the contour can be calculated with a following expression:

$$r = \sqrt{(cx - x)^2 + (cy - y)^2}, \quad (4)$$

where cx, cy are the coordinates of the center of mass; x, y are the coordinates of the point on the contour boundary.

In image recognition uses following notion of a descriptor, which denotes a unique one-dimensional discrete ordered sequence of numbers, which calculated on basic features of the object. Let's denote the descriptor as:

$$D = [d_1, d_2, ..., d_n], \quad (5)$$

where D is the descriptor; d is one of the numbers that form the descriptor; m is the number of elements of the descriptor.

For a contour consisting of n points, the number of Fourier coefficients is m = n / 2. We calculate separately real and imaginary parts of Fourier transformation, and then find the amplitude spectrum. The calculation of the Fourier coefficients begins with coefficients of the zero harmonic:

$$A_0 = \frac{1}{n}\sum_{i=1}^{n} r_i; \quad (6)$$

$$B_0 = 0, \quad (7)$$

where A, B are the real and imaginary parts of Fourier transformation; r is the distance from the center of mass to the point on the boundary; n is the total number of points that make up the contour; i is the counter. Members of the real part are calculated by the following formula:

$$A_0 = \frac{2}{n}\sum_{i=1}^{n} r_i \cos(\frac{2\pi ji}{n}), \quad (8)$$

After that, the remaining elements of the imaginary part B are found. The imaginary part is calculated by following formula:

$$B_j = \frac{2}{n}\sum_{j=1}^{n} r_j \cos(\frac{2\pi ji}{n}) \quad (9)$$

From coefficients A and B, an amplitude spectrum of C is:

$$C_j = \sqrt{A_j^2 + B_j^2} \quad (10)$$

Then the amplitude spectrum is normalized by zero harmonic, and the contour descriptor is defined as:

Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$D_{Fourier} = \left[\frac{|c_1|}{|c_0|}, \frac{|c_2|}{|c_0|}, \ldots, \frac{|c_n|}{|c_0|} \right], \qquad (11)$$

This descriptor is invariant to rotation and scaling. In addition, depending on the accuracy required, only the first few coefficients can be analyzed.

The final processing of images is binarization of every image by threshold. For RGB images a threshold is a color's value. There are a lot of situations where you need to select figures from a white sheet of paper as in Figure 4, so you will binarize them first. Each object of the image can be characterized by a set of certain features. The number of features depends on complexity of the object. The accuracy of the feature selection affects the efficiency of recognition of the object, which is described by the set. We consider image recognition based on a definite set of features, and mostly rely on Matlab Image Processing Toolbox.

Figure 4: Image binarization

We have some remarks concerning the quality of original images. In our case, original data is represented by a binary image. This simplifies our task at certain level, since the main emphasis in this example is on object recognition. However, in recognition of real images, in most cases, the important task is to convert an original image into a binary one. The quality of the conversion largely determines the quality of recognition.

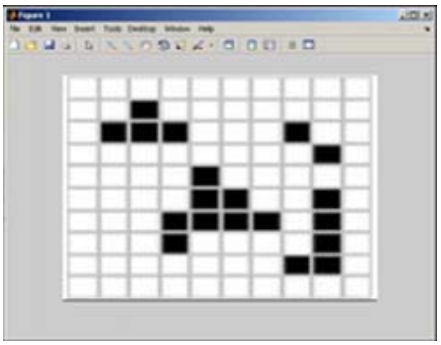

Figure 5: Converted image

In next example, we apply the most obvious statistical approach of classification of objects with morphometric features. The main morphometric features, which we have used, include a few shape features like solidity, extent, and etc. It is a complex and routine process to make right choice depending on particular situation. Since we use test images of simple form, we applied the 'extent' characteristic to the processing, which is defined by ratio of the scale of object to the selected box area. It is a value in the range [0,1]. Figure 6 shows the selection process based on the feature.

Figure 6: Box area selection

3. EXPERIMENTAL SETUP

Our goal is to create a neural network that can recognize images of correct word alignment. The solution has been implemented in MATLAB using Neural Network Toolbox.

First we need to decide how to submit data to the input of our network. The simplest and almost non-alternative solution is to express a two-dimensional image matrix to the one-dimensional vector. Particularly, for the image size of 28x28 we will have 784 entries, which is certainly not enough. Thereafter we choose architecture of the network.

Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

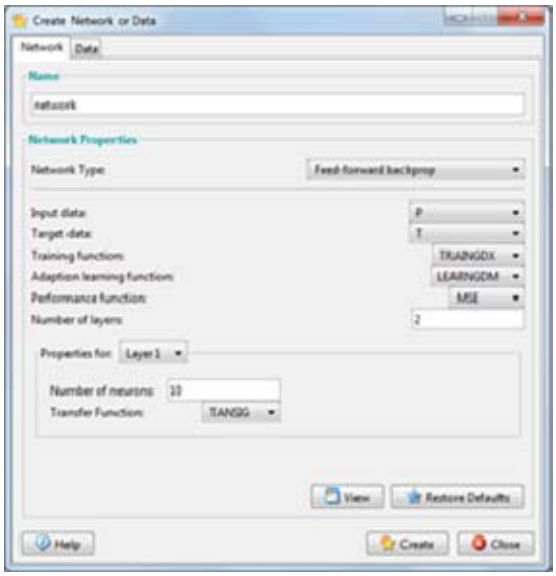

Figure 7: Initial window of neural network toolbox

Currently, there are no methods to clearly determine the structure and composition of the neural network based on description of a problem. We should say that for hard-to-formalize tasks, any one-shot method will hardly ever be created. In addition, there are many different methods of network optimization, as well as various heuristics and empirical rules. One such rule states that the number of neurons in a hidden layer must be at least in order of greater than the number of inputs. If we take into account that the transformation from an image into a class indicator is quite complex and non-linear problem, nobody can do it without errors.

As input, it is proposed to use a database of 50,000 training pairs of image-tag and 5000 test images without labels. All images are normalized by size and centered. At first, the number of neurons in hidden layers will be about 15,000 (10,000 in the 2nd layer and 5,000 in the third layer). For architectures with two hidden layers, the number of configurable and learning links are 10 million between the inputs and the first hidden layer, + 50 million between the first and second, and + 50,000 between the second and the output, assuming that we have 10 outputs, each of which means a number from 0 to 9. Totally, it is ~50,000,000 links.

The layers are customizable, which means that for each of them you will need to calculate the error gradient during learning. When we convert an image into a linear byte chain, we lose something irretrievably. And with each layer, this loss will only exacerbate. So, we may lose the topology of the image, i.e. the relationship between its individual parts. In addition, a task of image recognition implies the ability of neural networks to be resistant to small shifts, rotations and zooming.

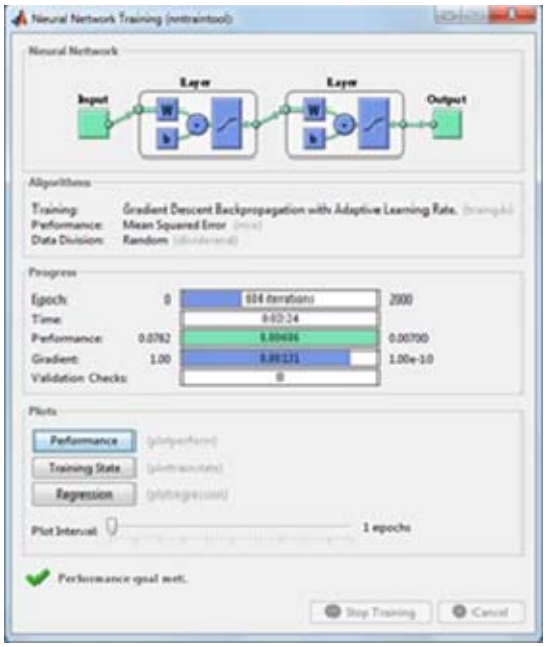

Figure 8 – A process window

At the training phase, the network converges after 10,000 epochs with a mean-square error of 0.001. The setup was tested 50 times on a test set of 50,000 images. The error rate is 0.1%. To test the efficiency of the algorithm, other training and testing datasets were created. The obtained results show that the algorithm allows recognizing complicated elements with high accuracy: for this data set after 30 tests we had 2 errors (the error rate was 0.15%). The experiment shows that 50+ neurons of the hidden layer can give the best result for the ratio of "learning time - convergence".

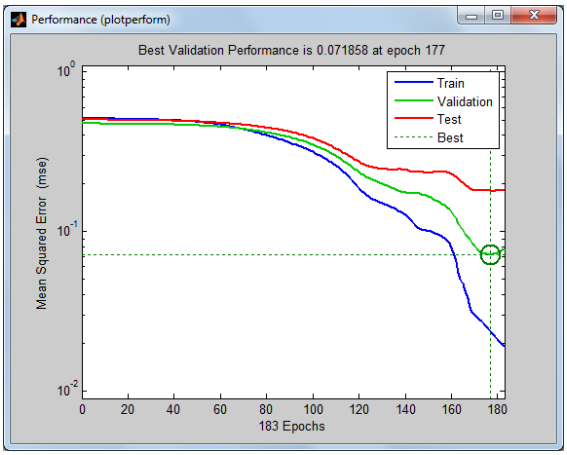

Figure 9: A network performance diagram

Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

We use various algorithms that require a real-time operation in MATLAB. This is a very handy package, the M-language of which allows you to focus on the algorithm itself, without worrying about allocating memory, operations, etc. In addition, many different tools allow you to create truly interdisciplinary applications in the shortest possible time. You can use Image Acquisition Toolbox to connect the webcam, use Image Processing Toolbox to process the image from it, use Neural Network Toolbox to generate and simulate models (Figure 8).

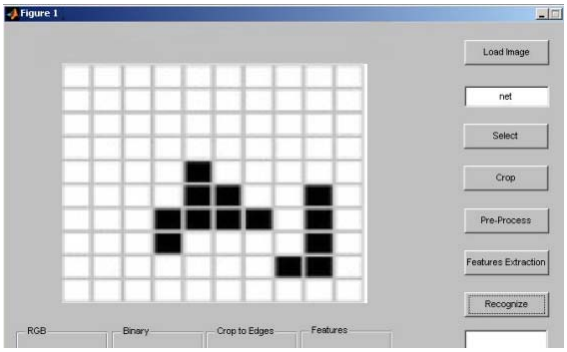

Figure 10: Image recognition GUI

Indeed, Neural Network Toolbox is a fairly versatile set of tools that allows you to create various neural networks of different types and architectures, but, unfortunately, not convolutional nets. The part of experiment was conducted in Python using the OpenCV package. It can create a dataset for learning, developing and training of neural networks and also has a separate interface for unit testing (Figure 10).

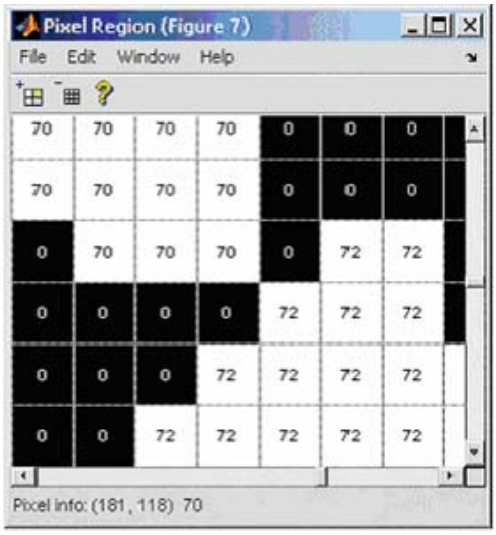

Figure 11: Normalized image file

3.1 Evaluation

We seek for better translation performance from at least two research directions. At first, let us discuss about the expected result of image classification. Before being able to solve image classification problems, a neural network must be trained. Its weights are tuned by a training algorithm, which takes a large set of training data to the input. Actually, we have an image file (Figure 11), and we would like to tag it automatically with a text label. The result of image classification algorithm should thus be a number, matching at certain level with contents of the file.

More precisely, the result of passing the image through the neural network (Figure 12) model trained with 100 classes will be a vector of 100 numbers, corresponding to the range of [0,1] associated to confidence. The "numbers" are related to a likelihood that if given result is a correct one. The likelihood should form some probability distribution. If the associated value with the image is 0.97, that means the algorithm has very high confidence on the image.

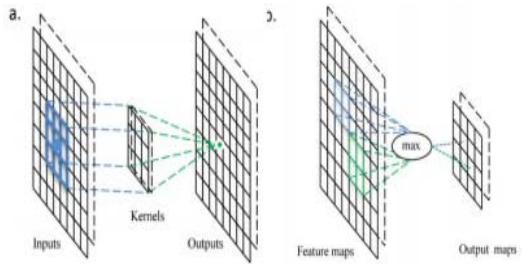

Figure 12: A neural network

Our application recognizes correct word alignment patterns of images by contour analysis using the Fourier descriptors and a neural network. We consider that the Fourier descriptors and neural networks are an effective mechanism for solving the problem of object recognition. The developed program is able to recognize complex shapes with high accuracy. To recognize the figures, a traditional multilayer neural network with back propagation has been applied. The regular sigmoid function was used as an activation function.

File access time is the time to read from the file, as this time is highly related with speed of the drive, rather than processor power. It is a bit interesting that it has the huge impact on total training time. At each training step, a large number of small files are processed in-memory. The optimization of disk access can produce significant

Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

reduction of this time. This result confirms that it needs to pre-calculate data usage in the training.

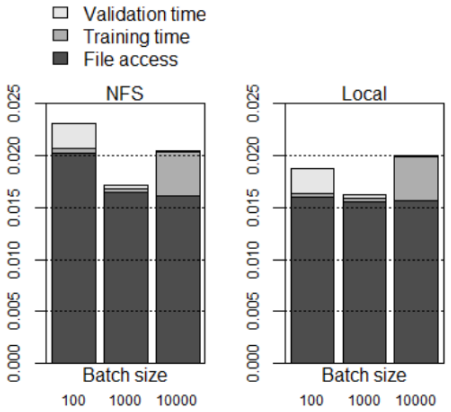

Figure 13: Network and local file systems speed

The essence of phrase-based models is a phrase table which comes from probability distributions. Most commonly, a phrase table is acquired from word alignments, which consists all phrases consistent with the alignment. Our studies try to reduce an impact of phrase table errors to the final translation performance based on different empirical statistics. In other words, we have made a study on the relationship between word alignments and translation quality from an experimental perspective proposed in this paper. The results show that we have to improve the quality of word alignment even if the system does not explicitly depend on the alignment.

In particular, low quality of alignment impacts to the translation at very minor level. This paper examines two extreme cases of alignment that are the best and the worst of word alignments to illustrate the situation. The results show that the quality of word alignment can significantly affect the translation measured by BLEU. Our work uses the baseline method to obtain a phrase table from word alignment, which trained by the EM (Expectation Maximization) algorithm. For phrase extraction from the word alignment, EM algorithm was applied to train the parallel corpus in several iterations, and then phrase pairs were extracted.

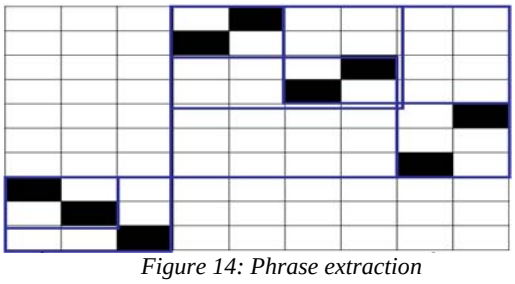

Figure 14: Phrase extraction

Current Phrase-Based SMT (PBSMT) systems[7] use the GIZA++ tool to produce word alignments, running the tool in both directions, source to target and target to source[8]. Then complex heuristics are applied to obtain a symmetrized alignment. For instance, a grow-diag-final method starts from the intersection of the two word alignments and enhance it with union alignment. We will not discuss this situation in this paper, where mostly the machine translation is surveyed. The size of phrase table is important for better word alignment, because the smaller size phrase table is more precise than big ones. Besides this, the machine translation performance depends on the size of the phrase table, since at least half of phrases can be removed without any loss in quality.

The quality of phrase extraction has a vital role in the overall translation quality. In the current phrase extraction algorithm phrases are directly injected to the phrase table, when a phrase is extracted from word alignment. In other words, word alignment affects the machine translation quality through phrase table.

In order to evaluate the experiments, we carried out some additional models using Moses toolkit [9], which provides a complete package for training the translation system. SMT requires language model (LM) to produce translations. Usually LM is created by the external toolkits like SRILM[10] or IRSTLM[11]. In our experiment, we use the IRSTLM toolkit for the large monolingual corpus. Before training, all the preprocessing was made using the integrated scripts in Moses. The language model is trained by n-gram model.

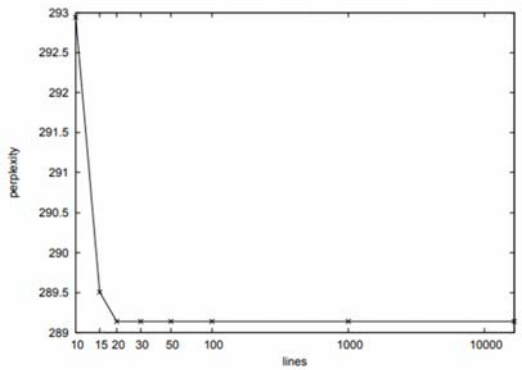

Figure 15: LM perplexity diagram

The length of the LM sequence is the n-gram order, depending on the value of the n, it is called unigram, bigram, and trigram model. The model is used to estimate how target sentences are enough grammatical due to the prior probability of the sequences. If they express a meaningful

Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

semantic relationship, they should be assigned to a higher likelihood by the language model; otherwise sequences should be assigned a lower likelihood.

Although we can improve the parameters of LM, due to data sparseness the majority of n-grams couldn't be observed in the training data. Despite of the fact that n-gram may be a quite grammatical and shows a high likelihood, the ML probability assigned to n-grams may be equal to zero. That may produce an unpredictable outcome.

We use interpolation and back-off for improving the reliability of word predictions in an LM. These techniques resolve the problem of data sparseness by the use of a low-order probability distribution.

Interpolation is a linear composition of higher-order and lower-order distributions. Actually, the less sparse low-order distributions are used to smooth the high-order distributions, resulting in that each n-gram model contributes a weight to the probability of the model. Interpolation presents a weighted sum of probabilities computed on a corpus of representative training data by using deleted estimation. These weights show the reliability of the distribution at each n-gram order on a held-out data set.

Language model smoothing merges an effect of the interpolation and back-off techniques to produce word sequence probabilities that are closer to the actual distribution of words. There are several smoothing methods, but we want to consider the Kneser-Ney smoothing method [12] as one of the most popular in modern SMT systems. A modified version of the method includes an interpolation procedure with lower-order distributions to improve LM perplexity (Figure 15). Perplexity is a positive number, by decreasing this value, the language model becomes better.

At the beginning of this experimental work we collected corpora from different domains. The corpus was not ready for immediate use and very raw. We have assembled news data on the Internet, also applied some normalization of data at the end of the collection phase to make it more suitable for model preparation. There are ~1,900,000 Kazakh sentences of news domain and more than 10 million English sentences. The statistics about the Kazakh corpora are shown in Table 1 and characteristics of the Kazakh language model is in Table 2.

Table 1: Monolingual corpora

Name	Domain	Sentences	Tokens
Zhasalas	news	~1,600K	53,520K
Adilet	legal	~240000	8,000,000
Akorda	news	~100000	3,500,000

Table 2: Language model

Tools	Size	Perplexity	Corpora
KenLM	521.2	296.51	Zhasalas
SRILM	117.7	289.92	Adilet
IRSTLM	78.6	271.20	Akorda

Our translation model based on the selected word alignment is a phrase-based translation model, prepared using Moses toolset with pre-processed data. Parallel data was aligned using a GIZA++ tool. Alignments are extracted in both directions, and then merged symmetrically using the grow-diagonal-final method. An n-gram model is used as the language model. Then, MERT [13] technique was applied to adjust the model parameters. The results of the translation were evaluated mostly by BLEU and few more metrics (Table 3).

Table 3: Translation performance scores

Translation model	Precision	Recall	F-score	AER	BLEU
Baseline	57.18	28.35	38.32	36.22	30.47
NN-IR*	71.12	28.31	42.49	20.19	31.9
Statistical	89.62	29.64	45.58	9.17	53.89

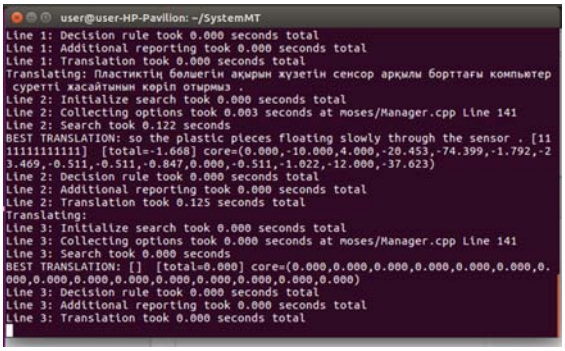

Figure 16: Translation output

The translation model estimates how well sentences in the foreign language are translations of sentences in the source language. Maximizing the reversed translation probability component tries to ensure that the output translation best corresponds semantically to the source sentence. While maximizing the language model component ensures that the generated translation is grammatically correct, fluent and commonly used. The process of finding this best translation is called decoding and it is performed by a component called the decoder. The language model, but then, measures how good a given hypothesis is written in the target language, literally, the fluency of generated sentences.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In early approaches our SMT systems have been built precisely on these two components, by factorizing each of them at the level of words. The search process is based on the argmax operation, and its later variation based on a log-linear model approach was proposed later by Och and Ney in 2003. It allows using more than two models and weighting them independently (Figure 16).
This allows us to consider additional features of the translation process. The most common features are the sentence-length, lexical and reordering models. Sentence-length model, usually called word penalty, compensates the system's tendency to prefer shorter translations for a better metric scores. Lexical models, also called as source-to-target and target-to-source models, are added in order to improve our reliance on a translation hypothesis. Reordering model represents how words have to move from their starting positions in the source sentence to their final positions in their translations for a pair of languages with a different word order [14].

Previously mentioned MERT algorithm consists of following steps: initial step, randomly or heuristically setting the weight values; translation, selecting n-best translation of the development set with current weights; comparing the MT score of the n-best translation with previous iteration; re-estimation, estimate new weights; iterate until feature weights have been converged. Because of the log-linear model we can use a few advantages over the noisy-channel model: basically, different weights for components of the model, easily adding new components to the model, usually called feature functions. This algorithm uses a held-out data set to maximize the translation quality according to certain automatic MT metric.

4. CONCLUSION

Accuracy and correctness are the most significant features in machine translation systems, which allow users more precisely operate text; efficiently comprehend its meaning, the most critical function of the system, do a better translation. We consider a set of methods to estimate correctness of Kazakh translation for a pair of languages with different word structure.

Our main assumptions are that the source text has been translated and relates to the target text context; also it is free from random errors. These methods are based on the features of phrases and dedicated to the estimation of every single text fragment. Because the methods are industrial standards in this field, we have conducted a set of experiments to evaluate the accuracy of application to the specific pair of languages. Basically, most of these methods are likelihood-based procedures, which estimates error rates in the certain sentence and make use of EM algorithm. These methods are applied to a sequence of translation outputs to measure a threshold between machine and human translation.

SMT systems came into existence half a century ago and have grown very rapidly. Today all major systems are practically identical for all translation subjects and contain huge databases of users experience in the digital era. Recently, a number of issues regarding mobile devices are raised, so we are focused on one of the fundamental issues, discovering properly translated fragments in a massive text.

We formally understand that the existing data sources don't have a strong relevance that we could use in evaluation, but generally we consider the main influence is not from database entries. Obviously, an accept rate of translation is the subject of well-known debates. In this case, we propose a brief introduction to the main principles of the translation evaluation. Our comparative evaluation of multiple translation outputs formed the experimental basis of this work, where we compared the translation outputs between distinct languages for their verbal constructions, word order, and grammar.

The main parts of our work based on the measure of overall correctness, but not on the fluency of translation. The coverage of the metrics highly depends on statistical reasons and a structure of the language, but intuitively we can make notions about the depth of connection between phrases, which may improve the overall accuracy. There are number of reasons why automatic evaluation should be considered in the general blueprint of experiments, exactly to distinguish translation errors from a properly processed output, especially for a post-editing or giving fair consideration about quality of the translation result. The process of selection valid sentences is time consuming, and needs a lot of resource, but for sure it will maximize a probability to produce more improved text.

This research is not dedicated to extend previous conceptual studies in MT evaluation with human judgment. We consider quality estimation as

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a measuring function of increasing the coverage and decreasing the number of errors in the target sentences. We take this term to express the probability that a given machine translation output and a reference human translation is similar in the certain sentence. The problems of errors that may randomly occur at the corpus preparation stages are not addressed here. There a lot of consequences of quality measuring that are out of scope of these methods.

Statistical machine translation often doesn't use linguistic information, fully relying on training data and statistical information for training and decoding. This approach is more convenient and language independent, which helps to quickly create MT systems for any language pair based on given corpora. However, in the real world parallel data is an expensive resource, and it is not always available in the enough quantity that is required for producing acceptable results. This problem is shown intensely for highly inflected languages and especially for certain domains. By combining some linguistic characteristics and the power of the statistical models, we probably can create more accurate translation systems.

But what are the flaws of machine translation and how exactly linguistic features can fulfill these gaps? First of all, it is ambiguity of natural languages. We cannot be sure of the meaning of some words and how they should be translated. Supplementing words with additional information may help solve their disambiguation and improve translation accuracy. The second problem is data sparseness. There are a lot of rare words and infrequent phrases in the training data that linguistic features can help generalize translation, and overcome the sparseness. Finally, machine translation is limited within an environment of the finite memory, processing time and corpora, over which it must operate. Special limits reducing performance are often placed to ensure that the training and decoding processes remain reliable. Machine translation results may get better if linguistically motivated constraints overcome these random limits.

Linguistic information does not guarantee a significant improvement of SMT, and in many cases, the integration of linguistic information may decrease overall performance. There are some reasonable explanations how it matters. Most of concerns are strongly correlated to the features of words, for instance, in Kazakh adjective-noun agreements need certain word classes, word morphology rules noun-phrase context. The non-standard phrase-based models must depend on linguistic properties and the surface form of words. Our study takes the advantage of making these properties work; because of they are usually more explicit, and it's easier to collect useful statistics of them.

As a research, SMT into Kazakh encounters the same challenges as any translation between any other language pairs, but has some specific issues closely related to the language, like rich agglutinative nature. It can be considered in special categories. There are a lot of languages with rich morphology like Kazakh, while others have a quite basic morphology. Natural languages can vary in by number of morphemes per word: In a few languages, each word has one morpheme like Chinese. These languages are known as isolating languages. While in some languages each word may have numerous morphemes. They are called agglutinative languages and there is a big challenge to segment the word into morphemes.

Kazakh is considered to be an agglutinative language. Like other Turkish languages, Kazakh language has a rich morphology with complex morphological inflections. A most of morphemes functions like English prepositions: cases, possessive nouns and verbs are usually affixed. The corpus of the Kazakh language is sparser than the similar English corpus in the fact that the number of found occurrences of Kazakh raw surface forms without morphological segmentation will be lower, than the number of the words in the English corpus.

This paper demonstrates that new technology of neural networks can be successfully applied to experimental data processing of natural language experiments. A key solution of our study is the system including complex data set for training of the hybrid neural network model and creation of validation methods. In future works, as our software becomes open source and well documented, its modular design will enable others to develop and extend the tool to easily add new features as required.

Our tool was designed with one main specific purpose, which is improving word alignments score for a large number of sentences, and evaluate them more efficiently. Custom software that enables the visualization of word alignment yet is going to be developed in future works and it will offer a tons of new features including online editing of alignments. We hold on

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the idea of using word alignments in an intuitive matrix style, but also make processing large volumes of data in a more straightforward way. In the future we aim to further enhance the software by making a number of additions. We plan to cover wider range of hardware platforms, including machines with different CPU and GPU types.

REFERENCES:

[1] P. Lambert, S. Petitrenaud, Y. Ma, and A. Way, "What types of word alignment improve statistical machine translation?", Machine Translation, vol. 26, no. 4, pp.289–323, 2012.

[2] Och F., Ney H. "A Systematic Comparison of Various Statistical Alignment Models", Journal Computational Linguistics, vol.29, no. 1, p.19-51, 2003.

[3] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: a method for automatic evaluation of machine translation", In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, 2002, pp. 311–318.

[4] C.T. Zahn, R.Z. Roskies, "Fourier descriptors for plane close curves", IEEE Trans. Computers, vol. 21, pp. 269-281, March 1972.

[5] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The Mathematics of statistical machine translation: parameter estimation," Computational Linguistics, vol. 19, no. 2, pp. 263–311, 1993.

[6] Jason Riesa. 2011. Picaro: A simple Command-Line Alignment Visualisation Tool. http://nlg.isi.edu/demos/picaro/

[7] Koehn P., "Statistical Machine Translation", Cambridge University Press, New York, 2009.

[8] F.J.Och and H.Ney,"A Comparison of Alignment Models for Statistical Machine Translation," in Proceedings of the 18th Conference on Computational Linguistics, vol.2, 2000, pp.1086–1090.

[9] Koehn P., Hoang H., Birch A., Callison-Burch C., Federico M., Bertoldi N., Cowan B., Shen W., Moran C., Zens R., Dyer C., Bojar O., Constantin A., Herbst E. "Moses: open source toolkit for statistical machine translation", In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Praha, 2007, p. 177-180.

[10] Stolcke A., "SRILM – An Extensible Language Modeling Toolkit", in Proc. Intl. Conf. Spoken Language Processing, Denver, 2002, pp. 257-286.

[11] Federico, M.; Bertoldi, N. and Cettolo, M., "IRSTLM: an open source toolkit for handling large scale language models", in INTERSPEECH-2008, 2008, pp. 1618-1621.

[12] Kneser R., Ney H. "Improved Backing-Off for n-gram Modeling", In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Berlin, 1995, pp. 181-184.

[13] Och F.J., "Minimum error rate training in statistical machine translation", In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, 2003, pp. 160-167.

[14] Bekbulatov E. and Kartbayev A., "A study of certain morphological structures of Kazakh and their impact on the machine translation quality", In Proceedings of the IEEE 8th International Conference on Application of Information and Communication Technologies, 2014, p.495-501.