Procesamiento del Lenguaje Natural, Revista nº 61, septiembre de 2018, pp. 15-22 recibido 20-03-2018 revisado 01-05-2018 aceptado 03-05-2018

Text-to-Pictogram Summarization for Augmentative and Alternative Communication

Resúmenes Texto-a-Pictograma para Comunicación Aumentativa y Alternativa

Laura Cabello, Eduardo Lleida, Javier Simón, Antonio Miguel, Alfonso Ortega
ViVoLab-Universidad de Zaragoza C/ María de Luna 1. 50018 Zaragoza
laura92cp2@gmail.com, (lleida,jasimon,amiguel,ortega)@unizar.es, http://vivolab.unizar.es

Abstract: Many people suffer from language disorders that affect their communicative capabilities. Augmentative and alternative communication devices assist learning process through graphical representation of common words. In this article, we present a complete text-to-pictogram system able to simplify complex texts and ease its comprehension with pictograms.
Keywords: pictogram, summarization, embeddings, augmentative and alternative communication, AAC, natural language processing, NLP

Resumen: Numerosas personas padecen trastornos del habla que merman su capacidad de comunicación. Su proceso de aprendizaje se apoya en el uso de dispositivos para la comunicación aumentativa y alternativa con símbolos gráficos. En este artículo presentamos un sistema texto-a-pictograma completo, capaz de simplificar textos complejos y facilitar su comprensión con pictogramas.
Palabras clave: pictograma, resumen, comunicación aumentativa y alternativa, CAA, procesado del lenguaje, PLN

1 Introduction

Texts are often illustrated because images help people to comprehend and remember the content. Moreover, graphic resources facilitate communication for individuals with severe language and speech disorders. Augmentative and alternative communication (AAC) encompasses all sort of communication methods that supplement and ease, or even replace, spoken and/or written language.

We present a novel text-to-pictogram system that aims to support AAC by conveying meaningful information to Spanish-speaking people with language impairments. Our system relies on pictograms or simple images to build sentences from a summarized text that captures the core meaning of an input document. Its main application is to serve as a reading aid to help understanding the theme from the input source and enhance learning process. Use case examples range from tales summaries for children to newspaper summaries for adults.

Since generating graphical replacement for different text genres is not an easy task, firstly we propose to summarize the input document to discard redundant information while retaining the core message. We are mainly concerned with what summary content should be regardless the form, so we define an extractive summarization method to extract salient sentences from the input source. Then, we retrieve a sequence of representative images for each word or constituent, i.e. group of words that function as a single unit within a hierarchical structure. This stage employs syntax information from part-of-speech (POS, from henceforth) labels, semantic features encoded in word embeddings, and context information from topic modeling. In order to improve understanding, we omit stopwords and further process complex text structures.

It is important to note that this work is not a study of summarization methods nor language modeling and feature learning techniques, but a complete text-to-pictogram system. The specific implementations presented here were chosen from literature after testing several options. Individually, they are simple but effective approaches to our ulti-

ISSN 1135-5948. DOI 10.26342/2018-61-1

mate objective of depicting a given article as a sequence of images.

2 Related Work

Previous research has been done in the field of text-to-pictogram systems. The current work differs from others in its nature, aiming to be an assistive tool to help people with language disorders understand complex texts. Also following same approach to extract pictures in a word by word basis, García-Cumbreras et al. (2016) presented an AAC system meant to be used in a controlled environment; Mihalcea and Leong (2006) studied a translation system through pictures; Zhu et al. (2007) rendered a nonlinear layout based on web search; UzZaman, Bigham, and Allen (2011) focused on multimodal summarization through images; Martínez-Santiago et al. (2015) proposes an upper ontology with linguistic knowledge used to model the usual language of beginning communicators. Another line of work is meant to represent a the gist of the text, known as text-to-scene systems. WordsEye (Coyne and Sproat, 2001) is one of the best known systems, able to synthesize realistic 3D scenes for descriptive sentences. **AraWord**[1] and **DictaPicto**[2] are examples for simple word by word basis translation to pictograms.

3 Method

Our text-to-pictogram system can be divided into three general phases, detailed in subsequent parts and depicted in Figure 2. Data preprocessing is a critical step in any Natural Language Processing (NLP) application. Thereafter, we use a sentence-ranking mechanism to form an extractive summary from an input document. Then, for each selected keyphrase, we exploit several NLP techniques -such as word embeddings and topic models-, deployed earlier at preprocessing stage to convert text to images. An output of our system can be seen in Figure 1.

3.1 Databases

Database **NEWS1709** was used to train word embeddings and LDA model. A custom web crawler searched through 32 different Spanish newspapers and magazines throughout 2017. We collected 850633 articles from January to September, covering a wide range of themes including but not limited to economy, politics, sports, forecast, culture and opinion. An initial preprocessing was required in order to get rid of advertisements, duplicate articles and non Spanish texts.

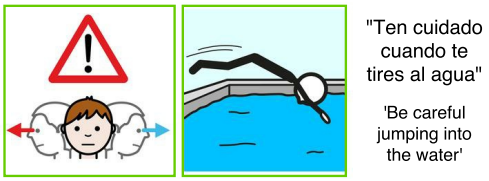

Figure 1: Example of a depicted sentence. Our system has detected the reflexive verb *'tirarse'* and output a single picture for the action *'tirarse al agua'* (*'jumping into water'*)

With regard to pictograms, **ARASAAC**[3] (Aragonese Portal of Augmentative and Alternative Communication) provides a package of 9564 color pictograms labeled in Spanish. Each pictogram is compose of three attributes: name, a set of labels or tags semantically related and the image itself. However, since some images can illustrate more than one concept, we are able to represent over 13000 different terms. This database is our main resource of images.

3.2 Preprocessing

Before directly tackling the problem of text-to-pictogram conversion, a preprocessing of our training data is required in order to define a voca-bulary enhanced with POS tags. We use the NEWS1709 database which is composed of raw text from newspaper articles, as explained in previous section. In addition, a set of word embeddings and a topic model are also created before being used in the process of selecting a suitable pictogram.

NLP tasks involve vast, not fixed vocabulary, since common expressions evolve and differ over time. Aiming to delimit our, we made use of a *lemmatizer* that removed inflectional endings to return the dictionary form of words, known as *lemma*. Concurrently, we gathered word's POS information such as category and type for those terms that appeared more than 100 times within the entire corpus. This way we filtered irrelevant terms and misspelling errors out and

[1] http://aulaabierta.arasaac.org/araword_inicio

[2] http://www.fundacionorange.es/aplicaciones/dictapicto-tea/

[3] www.arasaac.org

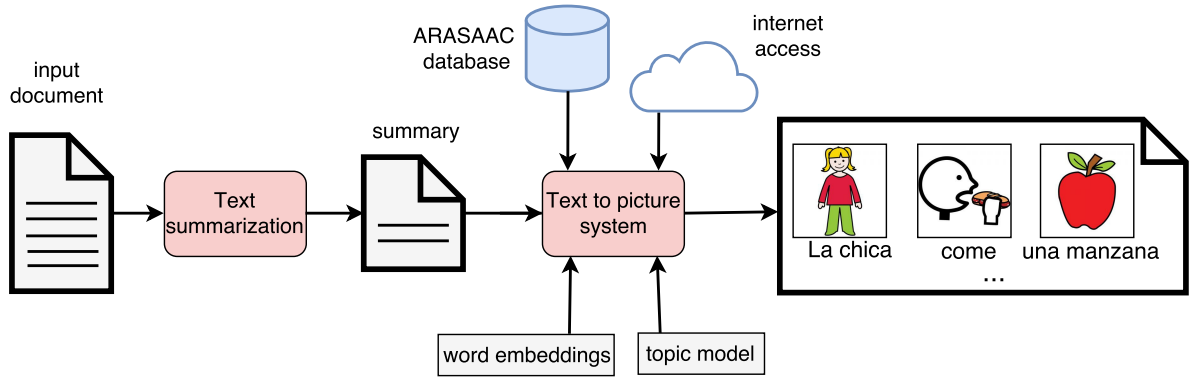

Figure 2: Block diagram depicting an overview of the end-to-end system

created a dictionary with 51358 terms. We stored 32 different labels after collecting all categories and types provided[4]. Both lemmatized content and word POS tags were achieved using the open-source suite FreeLing.

Word embeddings

Embeddings are a mathematic representation of words that captures a high degree of syntactic and semantic information. A word embedding $\vec{w}$ is a D-dimensional distributed representation of word w, from a vocabulary V, in a real valued vector space so as to $V \rightarrow \mathbb{R}^D : w \mapsto \vec{w}$.

Mikolov et al. (2013a) introduced word2vec, one of the first unsupervised embedding methods built upon a probabilistic prediction model, the continuous bag-of-words or CBOW. As shown in Section 4, we evaluated different approaches based on this model and observed that methods involving morphological information from POS tagging outperformed other implementations stu-died. Best results were provided by *wordtag* method, whose embeddings were trained over tagged text. For example, the sentence "el perro come" ("the dog eats") would be feed as "el/DA perro/NC come/VM", where DA: Determiner Article, NC: Noun Common, VM: Verb Modal.

Word embeddings are used in our end-to-end system in two scenarios. If the target word or constituent is not directly linked to a pictogram, we attempt to find a synonym from our vocabulary of pictograms defined by ARASAAC. We compute nearest neighbors to the target expression through the commonly used cosine similarity measure, defined as $sim(w_1, w_2) = \frac{\vec{w_1}\vec{w_2}}{||\vec{w_1}||||\vec{w_2}||}$. Another usage scenario applies if we're dealing with *polysemy*, this is, target expression potentially match more than one pictogram. In this case, word embeddings combined with a topic model are used to produce a sentence embedding, therefore allowing to retrieve the most suitable image as detailed in Section 3.4.

Topic modeling

Topic modeling provides methods for automatically organizing, searching or even understanding a large amount of data archives. Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan, 2003) is a popular method of topic modeling. It defines a generative probabilistic model of a corpus where each document is considered to exhibit multiple themes (or topics). Uncovering the hidden thematic structure from a considerable document collection allows us to generalize and subsequently classify unseen texts into predefined topics.

Despite topic modeling is inherent to extract representative content from documents, results strongly depend on the value given to hyperparameters α and η in Dirichlet distribution. Since we are working with a large, heterogeneous corpus, we tune α close to zero resembling a *mixture model* where only one topic is assigned per document: newspaper articles do not exhibit a mixture of many topics, but are more specific. $\alpha = 0.05$ draws a sparse probability density function (Θ_d), which means only few topics -usually one to three- will have positive probability within each document. Thus, expected distribution is not centered in the topic simplex. On the

[4]See manual of FreeLing for further information: https://talp-upc.gitbooks.io/freeling-user-manual/content/tagsets/tagset-es.html

other hand, η alters words distribution $\beta_{d,k}$. We set a high value ($\eta = 1.0$), so each topic is likely to contain a mixture of most of the words and not only a specific set of few words. This means all weights $w_d \in \beta_{d,k}$ are drawn from a probability distribution, which is important when solving the problem of *polysemy*.

Our proposal is to use LDA distribution of words over topics assigned to a document to weight words in a sentence. Therefore, most relevant words are given more importance and polysemic words are better disambiguated by context gathered in each topic.

Training of LDA model was performed over NEWS1709 dataset. Test and hold-out sets were created following same approach as in NEWS1709 but with press from October and November respectively. We monitored perplexity over test and hold-out corpora to ensure convergence and generalization of topics. In total, training represents 80% of the data, test and hold-out 10% each. Selected LDA model distinguishes $K = 50$ topics and it was trained with 10 passes over the entire corpus.

3.3 Text Summarization

Document summarization has been an active research area of NLP since the late 1950s (Luhn, 1958). Different approaches to make well formed summaries (Chang and Chien, 2009; Gong and Liu, 2001; Bian, Jiang, and Chen, 2014; Ozsoy, Alpaslan, and Cicekli, 2011) seek concise texts that convey important information in the original document(s). Most of them are extractive methods based on Latent Semantic Analysis (LSA) or LDA algorithms. We opted for a solution based on LSA over LDA due to the lack of necessity for previous training to tweak algorithm hyperparameters, which has a major impact in performance and often depends on external information such as the corpora used. Furthermore, it leads to a faster overall runtime (an average of 120s *vs* 2s in our tests).

LSA is an algebraic method applied to text summarization by Gong and Liu (2001). They applied singular value decomposition (SVD) to generic text summarization, and designed an unsupervised approach which does not need any previous training or outer information. After that, different LSA approaches have been proposed (Gong and Liu, 2001; Steinberger and Jezek, 2004; Murray, Renals, and Carletta, 2005; Ozsoy, Alpaslan, and Cicekli, 2011) which usually contain three common steps (Ozsoy, Alpaslan, and Cicekli, 2011):

i. *Define an input matrix.* Text must be mathematically readable. The process starts with creation of a term-sentence matrix $\mathbf{A} = [A_1, A_2, ..., A_n]$ whose columns define the importance of every word in each sentence. Cells can be filled in following different approaches, such as computing the word frequency in a sentence, the log-entropy value, or, as we did, the Tf-Idf (Term Frequency-Inverse Document Frecuency) if we count on a corpus with several documents. If there are a total of m terms and n sentences in the document, then we will have an mxn matrix $\mathbf{A}$ for the document, where without loss of generality, $m \geq n$. Since all words are not seen in all sentences, the matrix is usually sparse.

ii. *Apply SVD.* SVD is an algebraic method that decomposed the given matrix $\mathbf{A}$ into three new matrices, defined as follows:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V^T} \quad (1)$$

where $\mathbf{U}$ is an mxk column-orthonormal matrix containing k underlying concepts ($m > k$), $\mathbf{\Sigma}$ is a kxk diagonal matrix whose elements are non-negative singular values sorted in descending order, and $\mathbf{V^T}$ is a kxk orthonormal matrix whose columns are sentence singular vectors.

The interpretation of applying SVD to a term-sentence matrix is two-folded. From transformation point of view, it leads to a dimensionality reduction from an *m-dimensional* to a *k-dimensional* vector space. From semantic point of view, the SVD derives the latent semantic structure from the document represented by matrix $\mathbf{A}$ (Steinberger and Jezek, 2004).

iii. *Select salient sentences.* Different set of sentences are drawn depending on how we use the results from SVD. Several approaches solely use information within $\mathbf{V^T}$ matrix (Gong and Liu, 2001; Ozsoy, Cicekli, and Alpaslan, 2010), others also make use of $\mathbf{\Sigma}$ to emphasize most important concepts (Steinberger and Jezek, 2004; Murray, Renals, and Carletta, 2005). We implemented Topic method and Cross method proposed in (Ozsoy, Cicekli, and Alpaslan, 2010), and

chose the latter after comparing their performance. Cross method uses $\mathbf{V^T}$ matrix for sentence selection purposes.

3.4 Selecting Images

Once we have detected the main concepts from the input document, next stage is to find a sequence of pictograms or images to represent them. We retrieve a list of pictogram candidates from ARASAAC database and then call function in Algorithm 1 to get the most suitable image. If a word is out of ARASAAC dictionary of pictograms, we try to find a synonym computing cosine distance among word embeddings. Only in the event that a proper noun is not represented by ARASAAC pictograms, we do image search through a custom search RESTful API and get a public domain picture. Verbal periphrasis and compound verbal forms are depicted as a whole, making an emphasis in the main verb. Finally, stopwords are left without visual representation.

Our proposal to solve polysemy combines morphological information from POS tags, semantic information encoded in word embeddings and weights assigned to each word related to its importance within the K_d latent topics in the document, i.e., a hint of how relevant is *that* word in *that* document. Therefore, keywords are highlighted and words linked to more than one pictogram -typically polysemic words- are better distinguished. Let us consider the following example to endorse this statement. Note that it is an adaptation in English to ease reader's understanding. Suppose we are depicting the sentence 'Every year *cranes* return to the *wetlands*' from an article talking about wildlife, where keywords are in *italics*. Making use of our trained LDA model, two topics are assigned to it: *topic 1* related to nature and *topic 5* related to climate change, because the article also reads about it. Then, suppose we are searching for the best pictogram to represent the word *crane* (lemma from *cranes*) and we have reached *line 14* in *Algorithm* 1, which means there are potentially multiple suitable images.

Now, our approach to selecting the best image combines word embeddings and LDA topics. It consists of the following steps. First, we compute the sentence embedding $\vec{s}$ as a weighted sum of its L word embeddings. Word weights $\vec{z_s}$ take into account the topic-distribution in the document ($\Theta_{d,k}$ for topic k in document d) and term-distribution for each topic ($\beta_{d,k}$). Mathematically,

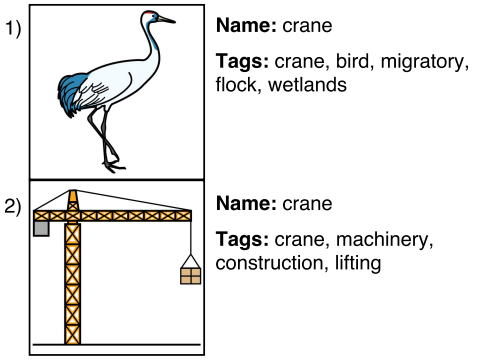

Figure 3: Retrieved pictos for the word *crane* from *ARASAAC* database

$$\vec{s} = \frac{\vec{z_s}}{||\vec{z_s}||}\mathbf{W}, \tag{2}$$

with

$$\vec{z_s} = \sum_{n=0}^{L-1} \sum_{k \in K_d} \Theta_{d,k}\beta_{d,k}z_{d,n}, \tag{3}$$

where $z_{d,n}$ is a binary variable that weights stopwords and words out of vocabulary by zero; $\mathbf{W}$ is an LxD matrix whose rows are word embeddings. The inner sum in Equation 3 looks at the topics from document d, i.e., topic 1 and 5 in the example above.

Next, we perform alike with tags attached to every image. Following our example, the word *cranes* have two different pictograms shown in Figure 3. Thus, we apply Equation 2 over topics from the original article to each set of tags and create $\vec{t_1}$ and $\vec{t_2}$. Word tags from picture 1 are closely related to words from original sentence and they are likely to have a higher weight in topics 1 and 5 than those from picture 2. This leads to a final embedding $\vec{t_1}$ that encodes a semantic akin to that in $\vec{s}$. Finally, we employ cosine similarity for similarity computations in the embedding space. Results in this particular example show that $sim(s, \vec{t_1}) > sim(s, \vec{t_2})$, so the right image is selected.

4 Evaluation

Successive evaluation metrics refer to different steps in our system pipeline. Before selecting the final implementation, we compared word embeddings with respect to specific queries and tested the quality of summaries given by different LSA approaches.

Algorithm 1 Select Picto

```
1: function SELECTPICTO(x, osent, lsent)
   Input x: list of picto candidates to represent the target word, osent: original sentence lsent: lemmatized sentence
   Output: selected picto
2:     if x.length = 1 then return x.picto
3:     context ← osent[w-W:w+W]
4:     initialize counter[] to zero
5:     for picto in x do
6:         if picto.name is compound and picto.name is in osent then
7:             return picto
8:         else
9:             if any word in context in picto.tags then
10:                counter[i] ← counter[i] + 1
11:    if max(counter) > 0 then
12:        return x[argmax(counter)]
13:    else
14:        picto ← word2vec.eval_context_with_LDA(x.tags, osent, lsent)
15:        return picto
```

Finally, we performed an overall test to objectively evaluate our text-to-pictogram system based on resources for AAC available in ARASAAC site.

4.1 Embeddings Evaluation

Table 1 displays a comparative among approaches proposed to enhance baseline word2vec performance with POS features. While in *wordtag* the POS is directly appended to training text as shown in Section 3.2, in *vectortag* it is encoded as one-hot vector and then concatenated to baseline word2vec embeddings. All embeddings mapped words into a 200-dimensional space, except *vectortag* that results in 200+32 dimensions, trained over 10 iterations with symmetric window of 4 samples and implemented negative sampling (Mikolov et al., 2013b) with 13 negative samples. Following previous work (Schnabel, Mimno, and Joachims, 2015; Mikolov, Yih, and Zweig, 2013), we conducted experiments on word analogy, relatedness and coherence tasks.

Word analogy assesses the capability of word embeddings to deduce semantic relationships. This task satisfies the statement *"if a:b, then x:y"* where y is unknown. Most suitable word y is found using word embeddings in the following function proposed by Mikolov, Yih, and Zweig (2013):

$$y^* = argmax_y sim(y, b) - sim(y, a) + sim(y, x) \quad (4)$$

We evaluate this metric on an Spanish version of Google analogy questions set proposed by Mikolov et al. (2013a).

Word relatedness and Coherence query inventories were created following work in (Schnabel, Mimno, and Joachims, 2015). We gathered 100 query words that balance frequency, POS (adjectives, adverbs, nouns and verbs) and concreteness. With regard to intrinsic evaluation of *relatedness*, the four nearest neighbors were retrieved for each of the 100 query words; 4 volunteers were then requested to pick the term that is most similar to the target word according to their perception. To evaluate *coherence*, we assess whether small, local clusters of words in the embedding space are mutually related. Same voluntary users were presented four words the day after, three of which are close neighbors (each query word from our 100 query words set and its two nearest neighbors) and one of which is an "intruder". Intruder word was selected to normalize frequency-based effects as in cited article.

	Analogy	Relatedness	Coherence
word2vec (baseline)	**49.4**	47.5 (67.0)	92.7
wordtag	49.3	**53.0 (75.5)**	**93.5**
vectortag	44.8	50.4 (70.7)	91.5

Table 1: Average accuracy scores (%) for each embedding method. Numbers between brackets show accuracy if acknowledging 2nd-nearest neighbor as valid. Best results for each metric are highlighted in bold

4.2 Summarization Evaluation

ROUGE evaluation tool (Lin, 2004) was adopted to measure summarization performance over a set of one hundred Spanish news extracted from NEWS1709 database. Articles were selected to have between 200 and 900 words and randomly belong to either one of the following categories: economy, international, national, society or sports. Golden standard summaries were provided by a linguistic expert. Table 2 contains different ROUGE evaluations that prove Cross

slightly better than Topic method, as presented in the original article. LSA algorithms were tuned so each summary covers up to half of the original length.

	LSA-Cross	LSA-Topic
ROUGE-1 R	0.7242	0.6913
ROUGE-1 P	0.5605	0.5611
ROUGE-1 F	0.6082	0.5840
ROUGE-L R	0.7015	0.6715
ROUGE-L P	0.5441	0.5450
ROUGE-L F	0.5900	0.5670
ROUGE-S R	0.5371	0.5138
ROUGE-S P	0.3453	0.3478
ROUGE-S F	0.3711	0.3470

Table 2: Results show a comparison of average recall (R), precision (P) and F-measure (F) for two different algorithms and ROUGE measures

ROUGE-1 measures unigram overlap between reference and automatic summaries, ROUGE-L measures the longest common subsequence of words and ROUGE-S scores overlap of word pairs (skip bigram) that can have an unlimited set of gaps in between words (Lin, 2004).

4.3 Text-to-pictogram Evaluation

It is not easy to devise an objective measure to quantify a text-to-pictogram system performance. In order to manage so, we counted on $N = 53$ sentences with 245 different lemmas and an average length of 10.5 words per sentence. Sentences have been extracted from two sources: an adapted document from the 2010 Convention on the Rights of Persons with Disabilities[5] and children's tales written by Douglas Wright[6], which were manually illustrated by ARASAAC. We adopted a binary evaluation and scored if a pictogram predicted by our system matched the one assigned by the illustrator. Since ARASAAC database includes different pictograms to depict exactly the same concept, we also counted them in as valid results. We achieved an averaged accuracy of 74% computed as $\frac{1}{N}\sum_i \frac{\#correct\ pictograms\ in\ sentence\ i}{\#total\ available\ pictograms\ in\ sentence\ i}$ where the denominator excludes some pictograms used but not included in ARASAAC database (public domain pictures, see Section 3.4).

5 Conclusion and Future Work

In this paper, we presented an assistive text-to-pictogram system to help people with language disorders to understand complex texts. The system integrates various phases combining automatic text summarization, natural language processing and a depicting algorithm that translates input text into images. We approached the issue of automatic summarization as a simple sentence ranking mechanism and then processed the summary with pre-trained embeddings and a topic model. Polysemy supposes a challenge in many aspects of NLP. We proposed to combine syntactic and morphological information from the local context of a polysemous word. Despite the fact that automatic illustration is an inherently subjective task, we conducted an overall test on children's book illustration and evaluated the different modules merged in our system. We realized the importance of a well defined evaluation method and corpus to foster the research in the area.

Our future compromise is to work on this line to provide a reference corpus for text-to-pictogram translation and summarization. Also we will concentrate on improving reading comprehension for natural language represented by pictograms alone by working jointly the ARASAAC professionals and experts on AAC systems at the Alborada [7] special education school in Zaragoza (Spain).

Acknowledgments

This work has been supported by the Spanish Government through project TIN2017-85854-C4-1-R and by the European Union's FP7 Marie Curie action, IAPP under grant agreement no.610986. We are grateful to ARASAAC for their determination to provide graphical resources for AAC. We are also thankful to our voluntary evaluators.

References

Bian, J., Z. Jiang, and Q. Chen. 2014. Research on multi-document summarization based on lda topic model. In *2014*

[5] http://www.ceapat.es/InterPresent2/groups/imserso/documents/binario/convencion_accesible2.pdf

[6] http://www.arasaac.org/materiales.php?id_material=578

[7] https://cpeealborada.wordpress.com/

Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics, volume 2, pages 113–116, August.

Blei, D., A. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. *Machine Learning Research*, 3:993–1022.

Chang, Y.-L. and J. T. Chien. 2009. Latent dirichlet learning for document summarization. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1689–1692, April.

Coyne, B. and R. Sproat. 2001. Toward communicating simple sentences using pictorial representations. In *Proceedings of 28th Conference on Computer Graphics and Interactive Techniques*, pages 487–496.

García-Cumbreras, M., F. Martínez-Santiago, A. Montejo-Ráez, M. Díaz-Galiano, and M. Vega. 2016. Pictogrammar, comunicación basada en pictogramas con conocimiento lingüístico. *Procesamiento del Lenguaje Natural*, 57:185–188.

Gong, Y. and X. Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25.

Lin, C. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out: Proceedings of ACL-04 Workshop*, pages 74–81.

Luhn, H. 1958. The automatic creation of literature abstracts. *IBM Journal of Research Development*, 2(2):159–165.

Martínez-Santiago, F., M. C. Díaz-Galiano, U. na López L. A., and R. Mitkov. 2015. A semantic grammar for beginning communicators. *Knowledge-Based Systems*, 86:158–172.

Mihalcea, R. and B. Leong. 2006. Toward communicating simple sentences using pictorial representations. In *Association of Machine Translation in the Americas*.

Mikolov, T., K. Chen, G. Corrado, and J. Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representations*, ICLR.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2 of *NIPS'13*, pages 3111–3119.

Mikolov, T., W.-T. Yih, and G. Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, HLT-NAACL, pages 746–751.

Murray, G., S. Renals, and J. Carletta. 2005. Extractive summarization of meeting recordings. In *6th Interspeech and 9th European Conference on Speech Communication and Technology*, pages 593–596.

Ozsoy, M., F. Alpaslan, and I. Cicekli. 2011. Text summarization using latent semantic analysis. In *Journal of Information Science*, volume 37, pages 405–417.

Ozsoy, M., I. Cicekli, and F. Alpaslan. 2010. Text summarization of turkish texts using latent semantic analysis. In *23rd International Conference on Computational Linguistics*, pages 869–876, August.

Schnabel, T., I. L. D. Mimno, and T. Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '15, pages 298–307.

Steinberger, J. and K. Jezek. 2004. Using latent semantic analysis in text summarization and summary evaluation. In *Proceedings of ISIM'04*, pages 93–100.

UzZaman, N., J. P. Bigham, and J. F. Allen. 2011. Multimodal summarization of complex sentences. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, pages 43–52.

Zhu, X., A. B. Goldberg, M. Eldawy, C. R. Dyer, and B. Strock. 2007. A text-to-picture synthesis system for augmenting communication. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, AAAI'07, pages 1590–1595.