Improving Text-to-Pictograph Translation Through Word Sense Disambiguation

Leen Sevens[*], Gilles Jacobs[**], Vincent Vandeghinste[*], Ineke Schuurman[*], Frank Van Eynde[*]
[*]Centre for Computational Linguistics (KU Leuven)
firstname@ccl.kuleuven.be
[**]Language and Translation Technology Team (Universiteit Gent)
gillesm.jacobs@ugent.be

Abstract

We describe the implementation of a Word Sense Disambiguation (WSD) tool in a Dutch Text-to-Pictograph translation system, which converts textual messages into sequences of pictographic images. The system is used in an online platform for Augmentative and Alternative Communication (AAC). In the original translation process, the appropriate sense of a word was not disambiguated before converting it into a pictograph. This often resulted in incorrect translations. The implementation of a WSD tool provides a better semantic understanding of the input messages.

1 Introduction

In today's digital age, people with Intellectual Disabilities (ID) often have trouble partaking in online activities such as email, chat, and social network websites. Not being able to access or use information technology is a major form of social exclusion. There is a dire need for digital communication interfaces that enable people with ID to contact one another.

Vandeghinste et al. (2015) are developing a Text-to-Pictograph and Pictograph-to-Text translation system for the WAI-NOT[1] communication platform. WAI-NOT is a Flemish non-profit organization that gives people with severe communication disabilities the opportunity to familiarize themselves with the Internet. Their safe website environment offers an email client that makes use of the Dutch pictograph translation solutions. The Text-to-Pictograph translation system (Vandeghinste et al., 2015; Sevens et al., 2015a) automatically augments written text with Beta[2] or Sclera[3] pictographs and is primarily conceived to improve the *comprehension* of textual content. The Pictograph-to-Text translation system (Sevens et al., 2015b) allows the user to insert a series of Beta or Sclera pictographs, automatically translating this image sequence into natural language text where possible. This facilitates the *construction* of textual content.

The Text-to-Pictograph translation process did not yet perform Word Sense Disambiguation (WSD) to select the appropriate sense of a word before converting it into a pictograph. Instead, the most frequent sense of the word was chosen. This sometimes resulted in incorrect pictograph translations (see Figure 1).

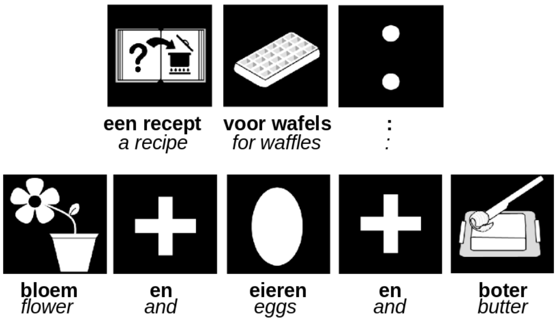

Figure 1: Example of Dutch-to-Sclera translation. The word *bloem* means both *flower* and *flour*. The most common sense is *flower*, which would be the wrong choice within the context of baking. Note that the pictograph language is a simplified language. Function words and number information are not represented.

We describe the implementation of a WSD tool

[1] http://www.wai-not.be/

[2] The Beta set consists of more than 3,000 coloured pictographs: https://www.betasymbols.com/

[3] Sclera pictographs are mainly black-and-white pictographs. Over 13,000 pictographs are available and more are added upon user request: http://www.sclera.be/

in the Dutch Text-to-Pictograph translation system. After a discussion of related work (section 2), we present both the Text-to-Pictograph translation tool and the WSD tool (section 3). We then proceed to describe the implementation procedure (section 4). Our evaluations show that improvements over the baseline in the Text-to-Pictograph translation tool were made (section 5). Finally, we conclude and describe future work (section 6).

2 Related work

There are not many works related to the task of translating text for pictograph-supported communication. Mihalcea and Leong (2008) describe a system for the automatic construction of simple pictographic sentences. They also use WordNet (Miller, 1995) as a lexical resource, but they do not use the WordNet relations between concepts in the same manner as the Text-to-Pictograph translation system does. Furthermore, their system does not translate the entire message. However, it should be noted that they make use of WSD in a way that is very similar to the approach described below. The WSD tool also relies on WordNet as a lexical database. Their system, though, is focused on English and the effectiveness of WSD within the context of a pictograph translation system was not evaluated.

Quite similar to the Text-to-Pictograph translation system are SymWriter[4] and Blissymbols (Hehner et al., 1983). These systems allow users to insert arbitrary text, which is then semi-automatically converted into pictographs. However, they do not provide automatic translation aids based on linguistic knowledge to properly disambiguate lexical ambiguities, which can lead to erroneous translation (Vandeghinste, 2012).

There is contradictory evidence that Natural Language Processing tools and Information Retrieval tasks benefit from WSD. Within the field of Machine Translation, Dagan and Itai (1994) and Vickrey et al. (2005) show that proper incorporation of WSD leads to an increase in translation performance for automatic translation systems. On the other hand, Carpuat and Wu (2005) argue that it is difficult, at the least, to use standard WSD models to obtain significant improvements to statistical Machine Translation systems, even when supervised WSD models are used. In later research, Carpuat and Wu (2007) and Chan et al. (2007) demonstrate that WSD can improve machine translation by using probabilistic methods that select the most likely translation phrase. Navigli (2009) underlines the general agreement that WSD needs to show its relevance in vivo. Full-fledged applications should be built including WSD either as an integrated or a pluggable component. As such, we set out to implement WSD and evaluate its effects within the Text-to-Pictograph translation system.

3 Description of the tools

The following sections describe the architecture of the Text-to-Pictograph translation system (section 3.1) and the WSD tool (section 3.2).

3.1 The Text-to-Pictograph translation system

The Text-to-Pictograph translation system translates text into a series of Beta or Sclera pictographs, cf. Vandeghinste et al. (2015) and Sevens et al. (2015a).

The source text first undergoes shallow linguistic processing, consisting of several sub-processes, such as tokenization, part-of-speech tagging, and lemmatization.

For each word in the source text, the system then returns all possible WordNet synsets identifiers (identifiers of sets of synonymous words) that are connected to that word. WordNets are an essential component of the Text-to-Pictograph translation system. For the Dutch system, Cornetto (Vossen et al., 2008; van der Vliet et al., 2010) was used. The synsets are filtered, keeping only those where the part-of-speech tag of the synset matches the part-of-speech tag of the word. Therefore, the semantic ambiguity of words across different grammatical categories (such as the noun *kom* 'bowl' and the verb *kom* 'come') has never formed an obstacle.

The WordNet synsets are used to connect pictographs to natural language text (see Figure 2). This greatly improves the lexical coverage of the system, as pictographs are connected to sets of words that have the same meaning, instead of just individual words. Additionally, if a synset is not covered by a pictograph, the links between synsets can be used to look for alternative pictographs with a similar meaning (such as the *dog* pictograph as a hyperonym for *poodle*). However, using pictographs through synset propagation (making use

[4] http://www.widgit.com/products/symwriter/

of the WordNet relations) is controlled by penalties for not using the proper concept.

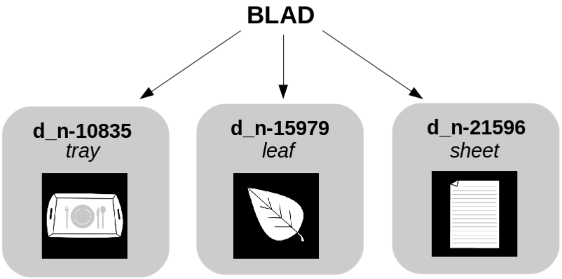

Figure 2: The Dutch word *blad* is linked to three different pictographs through its synsets.

Vandeghinste and Schuurman (2014) manually linked 5710 Sclera pictographs and 2760 Beta pictographs to synsets in Cornetto.

For every word in the sentence, the system checks whether one or more pictographs can be found for it. An A* algorithm[5] calculates the optimal pictograph sequence for the source text.

During the optimal path calculation step, the original system would sometimes be confronted with an equally likely choice between two or more pictographs, corresponding to different meanings of the same word (see Figure 2). In that case, the most commonly occurring sense according to DutchSemCor (Vossen et al., 2010) was chosen.

3.2 The Word Sense Disambiguation tool

We used the Dutch WSD tool that was made available by Ruben Izquiero[6] within the framework of the DutchSemCor project (Vossen et al., 2010).

DutchSemCor delivered a one-million word Dutch corpus that is fully sense-tagged with senses and domain names from the Cornetto database. It was constructed as a balanced-sense lexical sample for the 3000 most frequent and polysemous Dutch words, with about 100 examples for each sense. Part of the corpus was built semi-automatically and other parts manually. In the first phase, 25 examples were collected for each sense and manually tagged by annotators. The remainder of the corpus was tagged by a supervised WSD system, which was built using the manually tagged data from the first phase. The supervised system searched for the remaining 75 examples of the different senses to complete the corpus. Low-confidence examples were validated by annotators. In the last phase, even more examples were added to represent the context variety and the sense distribution as reflected in external corpora.

The resulting WSD system was built from the final sense-annotated corpus. The feature set that led to the best performance (81.62% token accuracy) contained words in a 1-token window around the target word, in combination with a bag-of-words representation of the context words. This WSD system takes natural language text as input and returns the confidence values of all senses according to Support Vector Machines.[7] Note that senses correspond to Cornetto synsets in both the Text-to-Pictograph translation tool and the WSD system.

4 Implementation

During the pre-processing phase, we let the Text-to-Pictograph translation system automatically assign a number to every sentence and every word. These numbers correspond to the sentences' position within the broader message and the words' position within the sentences. The WSD tool's output is numbered in a similar way. This way, if a particular input word appears multiple times within a message, the number label allows us to safely match that word with its correct WSD output counterpart.

The WSD tool is implemented after the shallow linguistic analysis and synset retrieval steps. The input to the WSD tool are the original sentences. Instead of only outputting one winning sense per word, we adapted the WSD tool to output the scores of each possible sense of the target word. As mentioned above, in the Text-to-Pictograph translation system, senses correspond to synsets which are attached to the word objects in the message. The WSD scores will now be added as a new feature of these synsets.

Next, we adapt the A* path-finding algorithm to include the WSD score in the penalty calculation as a bonus: A high WSD score biases the selection of the pictograph towards the winning sense. The score is weighted by a trainable parameter to determine the importance of WSD in relation to the

[5] A pathfinding algorithm that uses a heuristic to search the most likely paths first. Its input is the pictographically annotated source message, together with the pictographs penalties, depending on the number and kind of synset relations the system had to go through to connect them to the words.

[6] https://github.com/cltl/svm_wsd

[7] For a more detailed explanation on how the WSD system was built and tuned, we refer to Vossen et al. (2010).

Condition	BLEU	NIST	WER	PER
Beta				
No WSD	0.2572	5.0377	53.1435	45.5516
WSD	0.2721**	5.1976**	51.7200	43.7722
Sclera				
No WSD	0.1370	3.8321	72.1379	63.8621
WSD	0.1461*	3.9273	71.1724	62.8966

Table 1: Evaluation. $^{*}p < 0.05, ^{**}p < 0.01$

other system parameters.[8]

We have tuned these parameters through an automated procedure. The original tuning corpus consists of 50 messages from the WAI-NOT corpus, which were manually translated to Beta and Sclera pictographs by Vandeghinste et al. (2015). To the original tuning corpus, we added five more hand-picked messages from the corpus that included a polysemous word, that had at least two pictographs linked to at least two of its synsets. Biasing the tuning corpus like this was necessary, since the original set had very few ambiguous words.

We used the local hill climber algorithm as described in Vandeghinste et al. (2015), which varies the parameter values when running the Text-to-Pictograph translation script. The BLEU metric (Papineni et al., 2002) was used as an indicator of relative improvement. In order to maximize the BLEU score, we ran five trials of the local hill climbing algorithm, until BLEU converged onto a fixed score. Each trial was run with random initialization values, and varied the values between certain boundaries. From these trials, we took the best scoring parameter values.

5 Extrinsic evaluation

The evaluation set for the full Text-to-Pictograph translation system consists of 50 other messages from the WAI-NOT corpus, which were manually translated to Beta and Sclera pictographs by Vandeghinste et al. (2015).[9] We run the system with and without the WSD module. The system without WSD takes the most frequent sense for each word.[10] The automatic evaluation measures used are BLEU, NIST, Word Error Rate (WER) and Position-independent word Error Rate (PER).[11] We have added significance levels for the BLEU and NIST scores, by comparing the *no WSD* condition with the *WSD* condition. Significance was calculated using bootstrap resampling (Koehn, 2004).

The results are presented in Table 1.[12] Significant improvements were made for Beta and Sclera (in the BLEU condition). The observation that WSD does not more significantly improve the evaluation results can be explained by the fact that the evaluation set is small and does not contain many polysemous words with multiple senses which are linked to a pictograph in the evaluation set. Only six examples were found.

For that reason, we selected another 20 sentences from the WAI-NOT corpus that contain a word that has at least two pictographs attached to at least two of its synsets (belonging to the same grammatical category) and manually calculated the precision of their pictograph translations, focussing on the ambiguous words, before and after implementing the WSD tool. For Beta, choosing the most frequent sense for each word led to a correct translation for 14 out of 20 ambiguous words, while the addition of the WSD tool gave a correct translation for 18 out of 20 words. For Sclera, we get 11 out of 20 correct translations for the most frequent sense condition, and 17 out of 20 correct translations for the WSD condition. Looking back at Figure 1, the system will now correctly pick the flour pictograph instead of the flower pictograph within the context of baking.

6 Conclusion and future plans

We set out to implement and evaluate the effect of WSD on the Text-to-Pictograph translation system for the Dutch language. Improvements over the baseline system were made. We can affirm that disambiguation works in most cases where senses of ambiguous words are linked to pictographs in the lexical database. The system with WSD is now less likely to pick the wrong pictograph for an ambiguous word, effectively improving picto-

[8] See Vandeghinste et al. (2015) for an in-depth description of the other parameters.

[9] Creating a gold standard is difficult, as no parallel corpora are available. Translating the messages into Beta and Sclera pictographs is a meticulous and time-intensive process. This explains why the dataset is small.

[10] It is important to note that these two systems use two different sets of parameters for finding the optimal path as a result of separate parameter tuning.

[11] These metrics are used for measuring a Machine Translation output's closeness to one or more reference translations. We consider pictograph translation as a Machine Translation problem.

[12] The gap between the results for Sclera and the results for Beta is explained by Vandeghinste et al. (2015). The Sclera pictograph set consists of a much larger amount of pictographs than Beta, so several different paraphrasing reference translations are possible.

graphic communication for the end-users. Future work consists of implementing other WSD algorithms and enriching both the tuning corpus and the evaluation corpus with more expert reference translations of Dutch text into Beta and Sclera pictographs.

English and Spanish versions of the Text-to-Pictograph translation system are being developed.

Acknowledgments

We would like to thank IWT and the European Commissions Competitiveness and Innovation Programme for funding Leen Sevens doctoral research and the Able-To-Include project, which allows further development and valorisation of the tools. We also thank the people from WAI-NOT for their valuable feedback.

References

Marine Carpuat and Dekai Wu. 2005. Word Sense Disambiguation vs. Statistical Machine Translation. In *Proceedings of the 43th Annual Meeting on Association for Computational Linguistics*, pages 387–394. Association for Computational Linguistics.

Marine Carpuat and Dekai Wu. 2007. Improving Statistical Machine Translation Using Word Sense Disambiguation. *EMNLP-CoNLL*, 7:61–72.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word Sense Disambiguation Improves Statistical Machine Translation. *Annual Meeting - Association for Computational Linguistics*, 45:33.

Ido Dagan and Alon Itai. 1994. Word Sense Disambiguation Using a Second Language Monolingual Corpus. *Computational Linguistics*, 20(4):563–596.

Barbara Hehner, Peter A. Reich, Shirley McNaughton, and Jinny Storr. 1983. *Blissymbols for Use*. Blissymbolics Communication Institute.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of 2004 Conference on Empirical Methods on Natural Language Processing (EMNLP 2004)*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Rada Mihalcea and Chee Wee Leong. 2008. Toward Communicating Simple Sentences Using Pictorial Representations. *Machine Translation*, 22(3):153–173.

George Miller. 1995. WordNet: A Lexical Database fro English. *Communications of the ACM*, 28(11):39–41.

Roberto Navigli. 2009. Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, 41(2):30–35.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*, pages 311–318, Philadelphia, USA. Association for Computational Linguistics.

Leen Sevens, Vincent Vandeghinste, Ineke Schuurman, and Frank Van Eynde. 2015a. Extending a Dutch Text-to-Pictograph Converter to English and Spanish. In *Proceedings of 6th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT 2015)*, Dresden, Germany.

Leen Sevens, Vincent Vandeghinste, Ineke Schuurman, and Frank Van Eynde. 2015b. Natural Language Generation from Pictographs. In *Proceedings of 15th European Workshop on Natural Language Generation (ENLG 2015)*, pages 71–75, Brighton, UK. Association for Computational Linguistics.

Hennie van der Vliet, Isa Maks, Piek Vossen, and Roxane Segers. 2010. The Cornetto Database: Semantic issues in Linking Lexical Units and Synsets. In *Proceedings of the 14th EURALEX 2010 International Congress*, Leeuwarden, The Netherlands.

Vincent Vandeghinste and Ineke Schuurman. 2014. Linking Pictographs to Synsets: Sclera2Cornetto. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pages 3404–3410, Reykjavik, Iceland.

Vincent Vandeghinste, Ineke Schuurman, Leen Sevens, and Frank Van Eynde. 2015. Translating Text into Pictographs. *Natural Language Engineering*, pages 1–28.

Vincent Vandeghinste. 2012. Bridging the Gap between Pictographs and Natural Language. In *Proceedings of the RDWG Online Symposium on Easy-to-Read on the Web*.

David Vickrey, Luke Biewald, Marc Teyssier, and Daphne Koller. 2005. Word-Sense Disambiguation for Machine Translation. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 771–778. Association for Computational Linguistics.

Piek Vossen, Isa Maks, Roxane Segers, and Hennie van der Vliet. 2008. Integrating Lexical Units, Synsets, and Ontology in the Cornetto Database. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco.

Piek Vossen, Attila Grg, Ruben Izquierdo, and Antal Van den Bosch. 2010. DutchSemCor: Targeting the ideal sense-tagged corpus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.