Combination of Arabic Preprocessing Schemes for Statistical Machine Translation

Fatiha Sadat
Institute for Information Technology National Research Council of Canada
fatiha.sadat@cnrc-nrc.gc.ca

Nizar Habash
Center for Computational Learning Systems Columbia University
habash@cs.columbia.edu

Abstract

Statistical machine translation is quite robust when it comes to the choice of input representation. It only requires consistency between training and testing. As a result, there is a wide range of possible preprocessing choices for data used in statistical machine translation. This is even more so for morphologically rich languages such as Arabic. In this paper, we study the effect of different word-level preprocessing schemes for Arabic on the quality of phrase-based statistical machine translation. We also present and evaluate different methods for combining preprocessing schemes resulting in improved translation quality.

1 Introduction

Statistical machine translation (SMT) is quite robust when it comes to the choice of input representation. It only requires consistency between training and testing. As a result, there is a wide range of possible preprocessing choices for data used in SMT. This is even more so for morphologically rich languages such as Arabic. We use the term “preprocessing” to describe various input modifications applied to raw training and testing texts for SMT. Preprocessing includes different kinds of tokenization, stemming, part-of-speech (POS) tagging and lemmatization. The ultimate goal of preprocessing is to improve the quality of the SMT output by addressing issues such as sparsity in training data. We refer to a specific kind of preprocessing as a “scheme” and differentiate it from the “technique” used to obtain it. In a previous publication, we presented results describing six preprocessing schemes for Arabic (Habash and Sadat, 2006). These schemes were evaluated against three different techniques that vary in linguistic complexity; and across a learning curve of training sizes. Additionally, we reported on the effect of scheme/technique combination on genre variation between training and testing.

In this paper, we shift our attention to exploring and contrasting additional preprocessing schemes for Arabic and describing and evaluating different methods for combining them. We use a single technique throughout the experiments reported here. We show an improved MT performance when combining different schemes.

Similarly to Habash and Sadat (2006), the set of schemes we explore are all word-level. As such, we do not utilize any syntactic information. We define the word to be limited to written Modern Standard Arabic (MSA) strings separated by white space, punctuation and numbers.

Section 2 presents previous relevant research. Section 3 presents some relevant background on Arabic linguistics to motivate the schemes discussed in Section 4. Section 5 presents the tools and data sets used, along with the results of basic scheme experiments. Section 6 presents combination techniques and their results.

2 Previous Work

The anecdotal intuition in the field is that reduction of word sparsity often improves translation quality. This reduction can be achieved by increasing training data or via morphologically driven preprocessing (Goldwater and McClosky, 2005). Recent publications on the effect of morphology on SMT quality focused on morphologically rich languages such as German (Nießen and Ney, 2004); Spanish, Catalan, and Serbian (Popović...
and Ney, 2004); and Czech (Goldwater and McClosky, 2005). They all studied the effects of various kinds of tokenization, lemmatization and POS tagging and show a positive effect on SMT quality.

Specifically considering Arabic, Lee (2004) investigated the use of automatic alignment of POS tagged English and affix-stem segmented Arabic to determine appropriate tokenizations. Her results show that morphological preprocessing helps, but only for the smaller corpora. As size increases, the benefits diminish. Our results are comparable to hers in terms of BLEU score and consistent in terms of conclusions. Other research on preprocessing Arabic suggests that minimal preprocessing, such as splitting off the conjunctions, produces best results with very large training data (Och, 2005).

System combination for MT has also been investigated by different researchers. Approaches to combination generally either select one of the hypotheses produced by the different systems combined (Nomoto, 2004; Paul et al., 2005; Lee, 2005) or combine lattices/n-best lists from the different systems with different degrees of synthesis or mixing (Frederking and Nirenburg, 1994; Bangalore et al., 2001; Jayaraman and Lavie, 2005; Matusov et al., 2006). These different approaches use various translation and language models in addition to other models such as word matching, sentence and document alignment, system translation confidence, phrase translation lexicons, etc.

We extend on previous work by experimenting with a wider range of preprocessing schemes for Arabic and exploring their combination to produce better results.

3 Arabic Linguistic Issues

Arabic is a morphologically complex language with a large set of morphological features\(^1\). These features are realized using both concatenative morphology (affixes and stems) and templatic morphology (root and patterns). There is a variety of morphological and phonological adjustments that appear in word orthography and interact with orthographic variations. Next we discuss a subset of these issues that are necessary background for the later sections. We do not address derivational morphology (such as using roots as tokens) in this paper.

- **Orthographic Ambiguity:** The form of certain letters in Arabic script allows suboptimal orthographic variants of the same word to coexist in the same text. For example, variants of Hamzated Alif, \(\text{ا} > \text{ة} \) or \(\text{ة} < \) are often written without their Hamza (+). These variant spellings increase the ambiguity of words. The Arabic script employs diacritics for representing short vowels and doubled consonants. These diacritics are almost always absent in running text, which increases word ambiguity. We assume all of the text we are using is undiacritized.

- **Clitics:** Arabic has a set of attachable clitics to be distinguished from inflectional features such as gender, number, person, voice, aspect, etc. These clitics are written attached to the word and thus increase the ambiguity of alternative readings. We can classify three degrees of cliticization that are applicable to a word base in a strict order:

  \[\text{[CONJ+ [PART+ [Al+ BASE +PRON]]]}\]

At the deepest level, the BASE can have a definite article (+\(\text{ال} Al+ \) ‘the’) or a member of the class of pronominal enclitics, +PRON, (e.g. \(\text{هم} +hm \) ‘their/their’). Pronominal enclitics can attach to nouns (as possessives) or verbs and prepositions (as objects). The definite article doesn’t apply to verbs or prepositions. +PRON and Al+ cannot co-exist on nouns. Next comes the class of particle proclitics (PART+): +\(\text{ل} I+ \) ‘to/for’, +\(\text{ب} b+ \) ‘by/with’, +\(\text{k} k+ \) ‘as/such’ and +\(\text{s} s+ \) ‘will/future’. \(b+\) and \(k+\) are only nominal; \(s\) is only verbal and \(I+\) applies to both nouns and verbs. At the shallowest level of attachment we find the conjunctions (CONJ+) +\(\text{و} w+ \) ‘and’ and +\(\text{ف} f+ \) ‘so’. They can attach to everything.

- **Adjustment Rules:** Morphological features that are realized concatenatively (as opposed to templatically) are not always simply concatenated to a word base. Additional morphological, phonological and orthographic rules are applied to the word. An example of a morphological rule is the feminine morpheme, \(\text{s+p} (\text{ta marbuta})\), which can only be word final. In medial position, it is turned into \(\text{ت t}\). For example, \(\text{مكتبة} مكتبة mktbp+hm\) appears as \(\text{مكتبة} المكتبة mktbthm\) ‘their library’. An example of an orthographic rule is the deletion of the Alif (\(\text{ي} \)) of the definite article +\(\text{ال} Al+\) in nouns when preceded by the preposition +\(\text{ل} I+ \) ‘to/for’ but not with any other prepositional proclitic.

---

\(^1\)Arabic words have fourteen morphological features: POS, person, number, gender, voice, aspect, determiner proclitic, conjunctive proclitic, particle proclitic, pronominal enclitic, nominal case, nunation, idafa (possessed), and mood.
4 Arabic Preprocessing Schemes

Given Arabic morphological complexity, the number of possible preprocessing schemes is very large since any subset of morphological and orthographic features can be separated, deleted or normalized in various ways. To implement any preprocessing scheme, a preprocessing technique must be able to disambiguate amongst the possible analyses of a word, identify the features addressed by the scheme in the chosen analysis and process them as specified by the scheme. In this section we describe eleven different schemes.

4.1 Preprocessing Technique

We use the Buckwalter Arabic Morphological Analyzer (BAMA) (Buckwalter, 2002) to obtain possible word analyses. To select among these analyses, we use the Morphological Analysis and Disambiguation for Arabic (MADA) tool, an off-the-shelf resource for Arabic disambiguation (Habash and Rambow, 2005). Being a disambiguation system of morphology, not word sense, MADA sometimes produces ties for analyses with the same inflectional features but different lexemes (resolving such ties require word-sense disambiguation). We resolve these ties in a consistent arbitrary manner: first in a sorted list of analyses.

Producing a preprocessing scheme involves removing features from the word analysis and regenerating the word without the split-off features. The regeneration ensures that the generated form is appropriately normalized by addressing various morphotactics described in Section 3. The generation is completed using the off-the-shelf Arabic morphological generation system Aragen (Habash, 2004).

This preprocessing technique we use here is the best performer amongst other explored techniques presented in Habash and Sadat (2006).

4.2 Preprocessing Schemes

Table 1 exemplifies the effect of different schemes on the same sentence.

- **ST**: Simple Tokenization is the baseline preprocessing scheme. It is limited to splitting off punctuations and numbers from words. For example the last non-white-space string in the example sentence in Table 1, “trkyA.” is split into two tokens: “trkyA” and “.”. An example of splitting numbers from words is the case of the conjunction $+$ which can prefix numerals such as when a list of numbers is described: $15 \, w15$ ‘and $15’$.

$\text{The version of MADA used in this paper was trained on the Penn Arabic Treebank (PATB) part 1 (Maamouri et al., 2004).}$

Table 1 exemplifies the effect of different schemes on the same sentence.

- **ON**: Orthographic Normalization addresses the issue of sub-optimal spelling in Arabic. We use the Buckwalter answer undiacritized as the orthographically normalized form. An example of ON is the spelling of the last letter in the first and
fifth words in the example in Table 1 (wsynhY and AIY, respectively). Since orthographic normalization is tied to the use of MADA and BAMA, all of the schemes we use here are normalized.

• D1, D2, and D3: Decliticization (degree 1, 2 and 3) are schemes that split off clitics in the order described in Section 3. D1 splits off the class of conjunction clitics (w+ and f+). D2 is the same as D1 plus splitting off the class of particles (t+, k+, b+ and s+). Finally D3 splits off what D2 does in addition to the definite article AI+ and all pronominal enclitics. A pronominal clitic is represented as its feature representation to preserve its uniqueness. (See the third word in the example in Table 1.) This allows distinguishing between the possessive pronoun and object pronoun which often look similar.

• WA: Decliticizing the conjunction w+. This is the simplest tokenization used beyond ON. It is similar to D1, but without including f+. This is included to compare to evidence in its support as best preprocessing scheme for very large data (Och, 2005).

• TB: Arabic Treebank Tokenization. This is the same tokenization scheme used in the Arabic Treebank (Maamouri et al., 2004). This is similar to D3 but without the splitting off of the definite article AI+ or the future particle s+.

• MR: Morphemes. This scheme breaks up words into stem and affixival morphemes. It is identical to the initial tokenization used by Lee (2004).

• L1 and L2: Lexeme and POS. These reduce a word to its lexeme and a POS. L1 and L2 differ in the set of POS tags they use. L1 uses the simple POS tags advocated by Habash and Ram-
with the number of OOVs and perplexity. The only exceptions are L1 and L2, whose low OOV rate is the result of the reductionist nature of the scheme, which does not preserve morphological information.

5 Basic Scheme Experiments

We now describe the system and the data sets we used to conduct our experiments.

5.1 Portage

We use an off-the-shelf phrase-based SMT system, Portage (Sadat et al., 2005). For training, Portage uses IBM word alignment models (models 1 and 2) trained in both directions to extract phrase tables in a manner resembling (Koehn, 2004a). Trigram language models are implemented using the SRILM toolkit (Stolcke, 2002). Decoding weights are optimized using Och’s algorithm (Och, 2003) to set weights for the four components of the log-linear model: language model, phrase translation model, distortion model, and word-length feature. The weights are optimized over the BLEU metric (Papineni et al., 2001). The Portage decoder, Canoe, is a dynamic-programming beam search algorithm resembling the algorithm described in (Koehn, 2004a).

5.2 Experimental data

All of the training data we use is available from the Linguistic Data Consortium (LDC). We use an Arabic-English parallel corpus of about 5 million words for translation model training data.\(^3\) We created the English language model from the English side of the parallel corpus together with 116 million words the English Gigaword Corpus (LDC2005T12) and 128 million words from the English side of the UN Parallel corpus (LDC2004E13).\(^4\)

English preprocessing simply included lower-casing, separating punctuation from words and splitting off “”s”. The same preprocessing was used on the English data for all experiments. Only Arabic preprocessing was varied. Decoding weight optimization was done using a set of 200 sentences from the 2003 NIST MT evaluation test set (Mt03). We report results on the 2004 NIST MT evaluation test set (Mt04) The experiment design and choices of schemes and techniques were done independently of the test set. The data sets, Mt03 and Mt04, include one Arabic source and four English reference translations. We use the evaluation metric BLEU-4 (Papineni et al., 2001) although we are aware of its caveats (Callison-Burch et al., 2006).

5.3 Experimental Results

We conducted experiments with all schemes discussed in Section 4 with different training corpus sizes: 1%, 10%, 50% and 100%. The results of the experiments are summarized in Table 3. These results are not English case sensitive. All reported scores must have over 1.1% BLEU-4 difference to be significant at the 95% confidence level for 1% training. For all other training sizes, the difference must be over 1.7% BLEU-4. Error intervals were computed using bootstrap resampling (Koehn, 2004b).

Across different schemes, EN performs the best under scarce-resource condition; and D2 performs as best under large resource conditions. The results from the learning curve are consistent with previous published work on using morphological preprocessing for SMT: deeper morph analysis helps for small data sets, but the effect is diminished with more data. One interesting observation is that for our best performing system (D2), the BLEU score at 50% training (35.91) was higher than the baseline ST at 100% training data (34.59). This relationship is not consistent across the rest of the experiments. ON improves over the baseline

\(^3\) The parallel text includes Arabic News (LDC2004T17), eTIRR (LDC2004E72), English translation of Arabic Treebank (LDC2005E46), and Ummah (LDC2004T18).

\(^4\) The SRILM toolkit has a limit on the size of the training corpus. We selected portions of additional corpora using a heuristic that picks documents containing the word “Arab” only. The Language model created using this heuristic had a bigger improvement in BLEU score (more than 1% BLEU-4) than a randomly selected portion of equal size.

---

Table 2: Scheme Statistics

| Scheme | Tokens | OOVs | Perplexity |
|--------|--------|------|------------|
| ST     | 36000  | 1345 | 1164       |
| ON     | 36000  | 1212 | 944        |
| D1     | 38817  | 1016 | 582        |
| D2     | 40934  | 835  | 422        |
| D3     | 422085 | 575  | 137        |
| WA     | 38635  | 1044 | 596        |
| TB     | 42880  | 662  | 338        |
| MR     | 62410  | 409  | 69         |
| L1     | 36000  | 392  | 401        |
| L2     | 36000  | 432  | 460        |
| EN     | 55525  | 432  | 103        |
but only statistically significantly at the 1% level. The results for WA are generally similar to D1. This makes sense since $w^+$ is by far the most common of the two conjunctions D1 splits off. The TB scheme behaves similarly to D2, the best scheme we have. It outperformed D2 in few instances, but the difference were not statistically significant. L1 and L2 behaved similar to EN across the different training size. However, both were always worse than EN. Neither variant was consistently better than the other.

6 System Combination

The complementary variation in the behavior of different schemes under different resource size conditions motivated us to investigate system combination. The intuition is that even under large resource conditions, some words will occur very infrequently that the only way to model them is to use a technique that behaves well under poor resource conditions.

We conducted an oracle study into system combination. An oracle combination output was created by selecting for each input sentence the output with the highest sentence-level BLEU score. We recognize that since the brevity penalty in BLEU is applied globally, this score may not be the highest possible combination score. The oracle combination has a 24% improvement in BLEU score (from 37.1 in best system to 46.0) when combining all eleven schemes described in this paper. This shows that combining of output from all schemes has a large potential of improvement over all of the different systems and that the different schemes are complementary in some way.

In the rest of this section we describe two successful methods for system combination of different schemes: rescoring-only combination (ROC) and decoding-plus-rescoring combination (DRC). All of the experiments use the same training data, test data (MT04) and preprocessing schemes described in the previous section.

6.1 Rescoring-only Combination

This “shallow” approach rescores all the one-best outputs generated from separate scheme-specific systems and returns the top choice. Each scheme-specific system uses its own scheme-specific preprocessing, phrase-tables, and decoding weights. For rescoring, we use the following features:

- The four basic features used by the decoder: trigram language model, phrase translation model, distortion model, and word-length feature.
- IBM model 1 and IBM model 2 probabilities in both directions. We call the union of these two sets of features standard.
- The perplexity of the preprocessed source sentence (PPL) against a source language model as described in Section 4.3.
- The number of out-of-vocabulary words in the preprocessed source sentence (OOV).
- Length of the preprocessed source sentence (SL).
- An encoding of the specific scheme used (SC). We use a one-hot coding approach with 11 separate binary features, each corresponding to a specific scheme.

Optimization of the weights on the rescoring features is carried out using the same max-BLEU algorithm and the same development corpus described in Section 5.

Results of different sets of features with the ROC approach are presented in Table 4. Using standard features with all eleven schemes, we obtain a BLEU score of 34.87 – a significant drop from the best scheme system (D2, 37.10). Using different subsets of features or limiting the number of systems to the best four systems (D2, TB, D1 and WA), we get some improvements. The best results are obtained using all schemes with standard features plus perplexity and scheme coding. The improvements are small; however they are statistically significant (see Section 6.3).
6.2 Decoding-plus-Rescoring Combination

This “deep” approach allows the decoder to consult several different phrase tables, each generated using a different preprocessing scheme; just as with ROC, there is a subsequent rescoring stage. A problem with DRC is that the decoder we use can only cope with one format for the source sentence at a time. Thus, we are forced to designate a particular scheme as privileged when the system is carrying out decoding. The privileged preprocessing scheme will be the one applied to the source sentence. Obviously, words and phrases in the preprocessed source sentence will more frequently match the phrases in the privileged phrase table than in the non-privileged ones. Nevertheless, the decoder may still benefit from having access to all the tables. For each choice of a privileged scheme, optimization of log-linear weights is carried out (with the version of the development set preprocessed in the same privileged scheme).

The middle column of Table 5 shows the results for 1-best output from the decoder under different choices of the privileged scheme. The best-performing system in this column has as its privileged preprocessing scheme TB. The decoder for this system uses TB to preprocess the source sentence, but has access to a log-linear combination of information from all 11 preprocessing schemes.

The final column of Table 5 shows the results of rescoring the concatenation of the 1-best outputs from each of the combined systems. The rescoring features used are the same as those used for the ROC experiments. For rescoring, a privileged preprocessing scheme is chosen and applied to the development corpus. We chose TB for this (since it yielded the best result when chosen to be privileged at the decoding stage). Applied to 11 schemes, this yields the best result so far: 38.67 BLEU. Combining the 4 best pre-processing schemes (D2, TB, D1, WA) yielded a lower BLEU score (37.73). These results show that combining phrase tables from different schemes have a positive effect on MT performance.

6.3 Significance Test

We use bootstrap resampling to compute MT statistical significance as described in (Koehn, 2004a). The results are presented in Table 6. Comparing the 11 individual systems and the two combinations DRC and ROC shows that DRC is significantly better than the other systems – DRC got a max BLEU score in 100% of samples. When excluding DRC from the comparison set, ROC got max BLEU score in 97.7% of samples, while D2 and TB got max BLEU score in 2.2% and 0.1% of samples, respectively. The difference between ROC and D2 and ATB is statistically significant.

7 Conclusions and Future Work

We motivated, described and evaluated several preprocessing schemes for Arabic. The choice of a preprocessing scheme is related to the size of available training data. We also presented two techniques for scheme combination. Although the results we got are not as high as the oracle scores, they are statistically significant.

In the future, we plan to study additional scheme variants that our current results support as potentially helpful. We plan to include more
syntactic knowledge. We also plan to continue investigating combination techniques at the sentence and sub-sentence levels. We are especially interested in the relationship between alignment and decoding and the effect of preprocessing scheme on both.

Acknowledgments

This paper is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA. We thank Roland Kuhn and George Forster for helpful discussions and support.

References

S. Bangalore, G. Bordel, and G. Riccardi. 2001. Computing Consensus Translation from Multiple Machine Translation Systems. In Proc. of IEEE Automatic Speech Recognition and Understanding Workshop, Italy.

T. Buckwalter. 2002. Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, University of Pennsylvania. Catalog: LDC2002L49.

C. Callison-Burch, M. Osborne, and P. Koehn. 2006. Re-evaluating the Role of Bleu in Machine Translation Research. In Proc. of the European Chapter of the Association for Computational Linguistics (EACL), Trento, Italy.

M. Diab, K. Hacioglu, and D. Jurafsky. 2004. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL), Boston, MA.

R. Frederking and S. Nirenburg. 2005. Three Heads are Better Than One. In Proc. of Applied Natural Language Processing, Stuttgart, Germany.

S. Goldwater and D. McClosky. 2005. Improving Statistical MT through Morphological Analysis. In Proc. of Empirical Methods in Natural Language Processing (EMNLP), Vancouver, Canada.

N. Habash and O. Rambow. 2005. Tokenization, Morphological Analysis, and Part-Of-Speech Tagging for Arabic in One Fell Swoop. In Proc. of Association for Computational Linguistics (ACL), Ann Arbor, Michigan.

N. Habash and F. Sadat. 2006. Arabic Preprocessing Schemes for Statistical Machine Translation. In Proc. of NAACL. Brooklyn, New York.

N. Habash. 2004. Large Scale Lexeme-based Arabic Morphological Generation. In Proc. of Traitement Automatique du Langage Naturel (TALN). Fez, Morocco.

S. Jayaraman and A. Lavie. 2005. Multi-Engine Machine Translation Guided by Explicit Word Matching. In Proc. of the Association of Computational Linguistics (ACL), Ann Arbor, MI.

P. Koehn. 2004a. Pharaoh: a Beam Search Decoder for Phrase-based Statistical Machine Translation Models. In Proc. of the Association for Machine Translation in the Americas (AMTA).

P. Koehn. 2004b. Statistical Significance Tests for Machine Translation Evaluation. In Proc. of the EMNLP, Barcelona, Spain.

Y. Lee. 2004. Morphological Analysis for Statistical Machine Translation. In Proc. of NAACL, Boston, MA.

Y. Lee. 2005. IBM Statistical Machine Translation for Spoken Languages. In Proc. of International Workshop on Spoken Language Translation (IWSLT).

M. Maamouri, A. Bies, and T. Buckwalter. 2004. The Penn Arabic Treebank: Building a Large-scale Annotated Arabic Corpus. In Proc. of NEMLAR Conference on Arabic Language Resources and Tools, Cairo, Egypt.

E. Matusov, N. Ueffing, H. Ney 2006. Computing Consensus Translation from Multiple Machine Translation Systems Using Enhanced Hypotheses Alignment. In Proc. of EACL, Trento, Italy.

S. Nießen and H. Ney. 2004. Statistical Machine Translation with Scarce Resources Using Morphosyntactic Information. Computational Linguistics, 30(2).

T. Nomoto. 2004. Multi-Engine Machine Translation with Voted Language Model. In Proc. of ACL, Barcelona, Spain.

F. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In Proc. of the ACL, Sapporo, Japan.

F. Och. 2005. Google System Description for the 2005 Nist MT Evaluation. In MT Eval Workshop (unpublished talk).

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2001. Bleu: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176(W0109-022), IBM Research Division, Yorktown Heights, NY.

M. Paul, T. Doi, Y. Hwang, K. Imamura, H. Okuma, and E. Sumita. 2005. Nobody is Perfect: ATR’s Hybrid Approach to Spoken Language Translation. In Proc. of IWSLT.

M. Popović and H. Ney. 2004. Towards the Use of Word Stems and Sufﬁxes for Statistical Machine Translation. In Proc. of Language Resources and Evaluation (LREC), Lisbon, Portugal.

F. Sadat, H. Johnson, A. Agbago, G. Foster, R. Kuhn, J. Martin, and A. Tikuisis. 2005. Portage: A Phrase-based Machine Translation System. In Proceedings of the ACL Workshop on Building and Using Parallel Texts, Ann Arbor, Michigan.

A. Stolcke. 2002. Srlm - An Extensible Language Modeling Toolkit. In Proc. of International Conference on Spoken Language Processing.