IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 10, 2018, accepted June 25, 2018, date of publication June 28, 2018, date of current version July 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2851281

Effect of Word Sense Disambiguation on Neural Machine Translation: A Case Study in Korean

QUANG-PHUOC NGUYEN, ANH-DUNG VO, JOON-CHOUL SHIN, AND CHEOL-YOUNG OCK
Department of IT Convergence, University of Ulsan, Ulsan 44610, South Korea

Corresponding author: Cheol-Young Ock (okcy@ulsan.ac.kr)

This work was supported by the National Research Foundation of Korea through the Ministry of Education of the South Korea under Grant NRF-2016S1A5B6913773.

ABSTRACT With the advent of robust deep learning, neural machine translation (NMT) has achieved great progress and recently become the dominant paradigm in machine translation (MT). However, it is still confronted with the challenge of word ambiguities that force NMT to choose among several translation candidates that represent different senses of an input word. This research presents a case study using Korean word sense disambiguation (WSD) to improve NMT performance. First, we constructed a Korean lexical semantic network (LSN) as a large-scale lexical semantic knowledge base. Then, based on the Korean LSN, we built a Korean WSD preprocessor that can annotate the correct sense of Korean words in the training corpus. Finally, we conducted a series of translation experiments using Korean-English, Korean-French, Korean-Spanish, and Korean-Japanese language pairs. The experimental results show that our Korean WSD system can significantly improve the translation quality of NMT in terms of the BLEU, TER, and DLRATIO metrics. On average, it improved the precision by 2.94 BLEU points and improved translation error prevention by 4.04 TER points and 4.51 DLRATIO points for all the language pairs.

INDEX TERMS Lexical semantic network, neural machine translation, word sense disambiguation.

I. INTRODUCTION

MT systems that can translate text from one language to another have been a desire since the 1950s. Since then, various approaches have been investigated to build quality MT systems, such as dictionary-based, rule-based, example-based, statistics-based, and neural network-based. In the past two decades, statistics-based approaches have been used successfully to build MT systems. Currently, with the advent of robust deep learning, NMT has become the dominant paradigm in MT, with dramatic improvements compared with statistics-based approaches [1]–[3].

An NMT system first embeds each source word in a continuous vector (word embedding). Then, the system uses a recurrent neural network (RNN) to encode a source sentence (i.e., a sequence of word embeddings) into a single context vector [4], [5] or a sequence of them [6], [7]. Finally, the system uses another RNN to generate a target sentence. Because the continuous vector is encoded with multiple senses of a word, the encoder has an implicit weakness in WSD [8], [9]. Ambiguous words cause word choice problems, in which NMT systems must choose the correct target word from among several translation candidates that represent different senses of the input word.

Most languages contain many words with multiple senses or meanings. The sense of a word in a specific usage can only be determined by examining its context. For example, in the Korean sentence "*bae-leul meog-go bae-leul tass-deo-ni bae-ga a-pass-da*" (After eating a pear and getting on a boat, I had a stomachache), the word "*bae*" occurs three times and has three different meanings: a pear, a ship, and a stomach.

For this research, we built a Korean WSD preprocessor for NMT systems. The Korean WSD can annotate distinct sense-codes to homographic words using their particular context. The sense-codes are defined in the Standard Korean Language Dictionary (SKLD), as the representative of Korean homographic words. For instance, the sense-codes for the Korean word "*bae*" are defined from 01 to 12 to represent its 12 different senses. Because computers use blank spaces to separate words, tagging a sense-code to the word "*bae*" transforms it into a different word (e.g., bae_02). In this way, NMT systems can overcome the ambiguity of "*bae*."

Initially, we constructed an LSN named UWordMap for Korean. UWordMap consists of a noun network and a predicate network with a hierarchical structure for hyponymy relations. The noun and verb networks are connected through subcategorization information. To the best of our knowledge, UWordMap is currently the biggest and most comprehensive Korean LSN, containing nearly 500,000 nouns, verbs, adjectives, and adverbs. We then applied UWordMap to build a fast and accurate Korean WSD system.[1]

We conducted a series of bi-directional translation experiments with Korean-English, Korean-French, Korean-Spanish, and Korean-Japanese language pairs. The experimental results show that our Korean WSD system can significantly improve NMT translation quality in terms of the BLEU, TER, and DLRATIO metrics. On average, it improved precision by 9.68 and 5.46 BLEU points for translation from and to Korean, respectively. It also improved translation error prevention by 8.9 TER points and 8.0 DLRATIO points for all the tested systems.

II. RELATED WORK

Early research tried to prove that WSD could benefit MT, but Carpuat and Wu [10] reported negative results from integrating a Chinese WSD system into a Chinese-to-English word-based statistical MT (SMT) system. Their WSD predicted Chinese word senses using the HowNet dictionary and then projected the predicted senses into English glosses.

Instead of predicting the senses of ambiguous source words, Vickrey *et al.* [11] reformulated the WSD task for SMT as predicting possible target translations. Carpuat and Wu [12] integrated multi-word phrasal WSD models into a phrase-based SMT. Their experiments both led to the conclusion that WSD can improve SMT.

Following those successful integrations of WSD into SMT, others considered applying WSD systems to MT using several methods. Xiong and Zhang [13] proposed a sense-based SMT model. Su *et al.* [14] used a graph-based framework for collective lexical selection. They both experimented on Chinese-to-English translations and achieved improvements in translation quality.

In addition to Chinese-English translations, WSD systems have been successfully integrated into translations of other language pairs. Using word senses as contextual features in maxent-based models enhanced the quality of an English-Portuguese SMT [15]. A Czech-English phrase-based SMT was improved using a verb-only WSD [16]. A WordNet-based WSD was successful in an English-Slovene SMT [17].

Recently, Rios *et al.* [18] proposed a method to improve WSD in NMT by adding sense to word embeddings and extracting lexical semantic chains from the training data. Liu *et al.* [19] also added context-aware to word embeddings. In their experimental results, the NMT system failed to translate ambiguous words, and their WSD improved the quality of the translation results. Both methods customized translation models to learn additional information, which might lead to low performance. In particular, increasing the size of the training corpus and using more deep layers could cause performance to diminish exponentially.

In contrast to the previous research, we did not modify the NMT model. Instead, we propose a fast and accurate WSD system that can run independently. Our WSD acts as a preprocessor to annotate Korean texts with sense-codes before they are input into the NMT system. The sense-codes are not additional information; instead, they transform ambiguous words. Tagging a single word with different sense-codes creates new words, and consequently removes ambiguos words.

III. UWordMap – A KOREAN LSN

Because an LSN is used as an essential and useful knowledge resource in various natural language processing systems, especially in systems dealing with semantics, many researchers have tried to construct one for each language; examples include the Princeton WordNet [20] for English, EuroWordNet [21] and BalkaNet [22] for various European languages, and HowNet [23] for Chinese. Several projects have been conducted to build a Korean LSN, but most of them are based on existing non-Korean LSNs. KorLex [24] and the Korean Thesaurus [25] were based on WordNet, and CoreNet [26] was developed by mapping the NTT Goidaikei Japanese hierarchical lexical system to Korean word senses. Some Korean LSNs were designed for specific tasks; for instance, the ETRI lexical concept network [27] was designed for question-answering systems.

The Korean LSN UWordMap was manually constructed as a large-scale lexical semantic knowledge base. In UWordMap, every node corresponds to a certain sense and has a unique sense-code that represents each distinct sense of its associated lexicon. The lexicons and their sense-codes were extracted from SKLD.

To construct UWordMap, we first established a lexical network for nouns; it has a hierarchical structure for hyponymy relations. Then, we constructed a lexical network for predicates, and it not only has a hierarchical structure for hyponymy relations but also provides subcategorization information to connect each predicate with the lexical network for nouns. Furthermore, we also defined combination relations for adverbs and predicates.

UWordMap can be used through its application-programming interface [28] or our online service.[2] In this research, we used only the lexical network for nouns and the subcategorization information for predicates to improve the accuracy of our WSD systems. Hence, we next describe the structure of the lexical network for nouns and the subcategorization information for predicates, which are shown in FIGURE 1.

[1] http://nlplab.ulsan.ac.kr/doku.php > UTagger

[2] http://nlplab.ulsan.ac.kr/doku.php?id=uwordmap

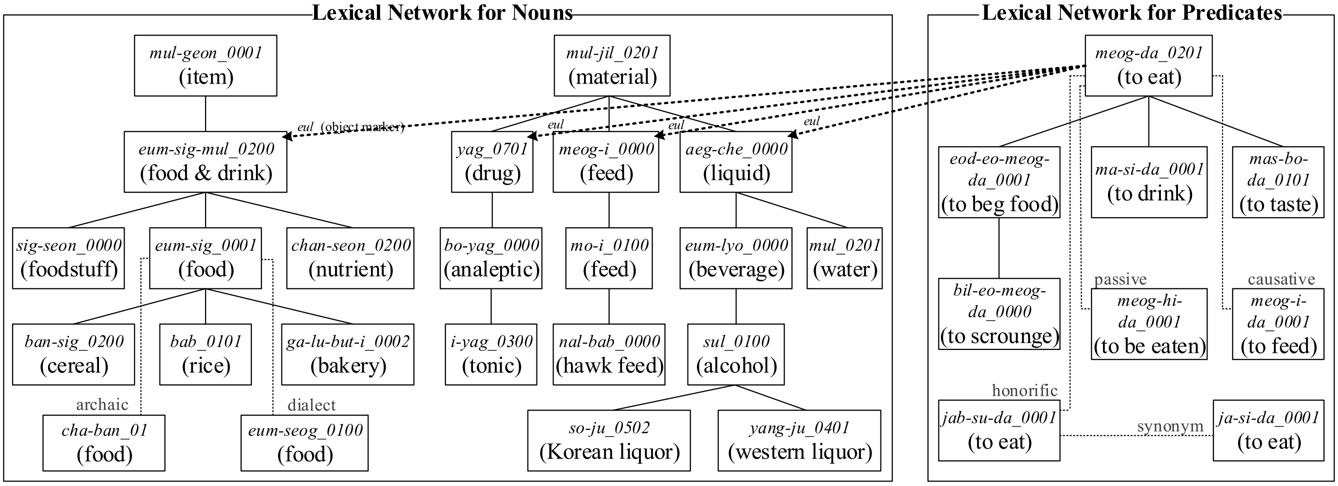

FIGURE 1. Overview of lexical network for nouns and predicates in UWordMap.

A. LEXICAL NETWORK FOR NOUNS

The lexical network for nouns (LNN) was designed as a hierarchical structure network, in which an upper-level node is a hypernym of lower-level nodes. Each node is connected to only one upper-level node and to one or more lower-level nodes through hyponymy relations. In other words, an LNN node cannot have multiple hypernyms.

In addition to the hyponymy relation, the LNN contains other relations between nodes: absolute synonymy (same meaning), partial synonymy (similar meaning), antonymy, and association relations, which are shown in FIGURE 1 using dashed lines.

We constructed the LNN using the following steps.

1) DETERMINE A SET OF TOP-LEVEL NODES

Top-level nodes have no upper-level node. The set of top-level nodes is the basic frame of every hierarchical structure network. Determining the set of top-level nodes is thus the most important step in constructing an LNN, which needs to be a balanced and expandable network. We used the following principles to select our set of top-level nodes.

- Top-level nodes must be lexicons registered in SKLD.
- Top-level nodes must have clear meanings.
- Top-level nodes must be used frequently.
- No top-level node may share any duplicate concept with another top-level node.
- The selection of top-level nodes should consider the composition of the lower-level nodes.

Using those principles, we determined the set of 23 top-level nodes shown in Table 1.

TABLE 1. Top-level nodes of lexical network for nouns.

Top-level	Meaning	Direct Hyponyms
gong-gan_0502	space	gos_0101(place), ji-yeog_0302(area)...
gwa-jeong_0300	process	dan-gye_0300(step), sa_0801(chronicle)...
gwan-gye_0501	relation	sun-seo_0001(order), yu-dae_0200(tie)...
gi-ho_1000	symbol	do-hyeong_0302(figure)...
dan-wi_0201	unit	hwa-pye-dan-wi_0000(currency unit)...
dae-sang_1101	object	mog-pyo_0001(target) ...
mo-yang_0201	shape	oe-yang_0300(appearance)...
mul-geon_0001	item	go-mul_0602(debris), gi-gi_1300(device)...
bang-beob_0001	method	bang-sig_0100(way)...
beom-wi_0001	scope	bun-ya_0001(field), bu-bun_0100(part)...
saeng-mul_0101	organism	dong-mul_0001(animal)...
seong-jil_0002	characteristic	bon-jil_0201 (nature)...
si-gan_0401	time	dong-an _0101(while) ...
yo-so_0401	element	seong-bun_0104(constituent)...
in-ji_0801	cognition	saeng-gag_0101(thought)...
jag-yong_0101	effect	yeong-hyang_0400(impact)...
jae-lyo_0101	material	gam_0202(cloth) ...
jeong-do_1101	degree	gan-gyeog_0203(distance)...
jon-jae_0001	existence	sil-jae_0201 (reality)...
jong-lyu_0201	kind, type	in-jong_0102(race)...
jib-dan_0000	organization	sa-hoe_0701(society)...
haeng-wi_0001	action	gae-bal_0001(development)...
him_0103	power	gwon-wi_0001(authority)...

2) HYPONYMY RELATION ESTABLISHMENT

The hyponymy relation is the core of the LNN: lower-level nodes have an IS-A relation to their upper-level nodes. We used both top-down and bottom-up strategies to establish the hyponymy relations based on the following principles.

- Upper-level nodes and lower-level nodes are connected by their lexical semantics.
- Upper-level nodes contain information about their lower-level nodes.
- Lower-level nodes inherit the properties of their upper-level nodes.
- For single nouns or suffixes that originate from Chinese characters, the relation relies on the meaning of those Chinese characters. For instance, the words "*bal-jeon-so*" (power plant) and "*sa-mu-so*" (office)" are connected to the upper-level node "*jang-so*" (place) through the suffix "*so*" (place).
- For compound nouns, the relation relies on the right side component, which usually contains the core meaning.

- For terminologies, the relation relies on existing terminological classification systems.

After establishing the hyponymy relations, we used the lexical semantic model of Cruse [29] to examine the LNN and ensure that its structure observes the IS-A relation.

3) SYNONYMY RELATION ESTABLISHMENT

Synonymy relations in the LNN are classified into two categories: absolute synonymy relation (ASR) and partial synonymy relation (PSR). The ASR applies when two or more words have the same meaning, and the PSR connects words with similar meanings. For more detail, ASR is divided into six types: standard absolute synonymy, misused words, dialect words, North Korean words, archaic words, and short form–original form words. PSR is divided into eight types: standard partial synonymy, refining words, aspirated sound words, honorific words, familiar speech, jargon, terminology, and designation words.

4) ANTONYM RELATION ESTABLISHMENT

We established the antonym relation for word pairs with opposite meanings. According to the lexical semantics, we divided the antonym relation into three kinds: complementary antonym, gradable antonym, and relative antonym.

B. LEXICAL NETWORK FOR PREDICATES

Korean verbs and adjectives have similar grammatical constructions. Korean grammar does not require a verb "to be" (e.g., am, are, is) for an adjective to construct a sentence. Korean adjectives thus function as stative verbs that play the predicate role in sentences [30]. Hence, we arranged verbs and adjectives into a single lexical network for predicates (LNP).

In addition to having a hierarchical structure for hyponymy relations, the LNP also contains subcategorization information about syntactic categories that is construed as arguments. Each predicate associates with its arguments in forming a predicate-argument structure that we used to define connections between predicates and the least common subsumers (LCS) of the LNN. Subcategorization information is essential for generating semantic connections between the LNN and LNP in UWordMap.

We constructed subcategorization information for all predicates registered in SKLD by extracting the arguments from example sentences of each predicate in SKLD. Based on the arguments, we connected the predicates with the LCS of the LNN. For instance, in FIGURE 1, the predicate "*meog-da*" (to eat) was connected to LCS such as "*eum-sig-mul*" (food), "*yag*" (drug), "*meog-i*" (feed), and "*aeg-che*" (liquid) through the sentence pattern "*eul*" (object case marker). However, some hyponyms of "*yag*" cannot be connected to the verb "meog-da" (i.e., "*ba-leu-neun-yag*" (liniment), "*ju-sa-yag*" (injection), and "*but-i-neun-yag*" (medicated plaster)). We marked those cases with a constrained relation [N_OBJ].

The principles for constructing the subcategorization information for predicates are as follows.

1- Refer to the example sentences of each predicate in SKLD to construct predicate-argument structures. For instance, from the examples of the predicate "*meog-da_0201*" (to eat) in SKLD, we extracted the sentence pattern "*eul*" and combinative argument nouns: {*bab* (rice), *sul* (alcohol), *yag* (drug), *mul* (water), *eum-sig* (food), *mo-i* (feed), *bo-yag* (analeptic)}.

2- Connect the predicate with the argument noun's upper-level nodes in the LNN. For instance, in the LNN, the upper-level nodes of those argument nouns are {*eum-sig* (food), *eum-lyo* (drink), *mul-jil* (material), *aeg-che* (liquid), *eum-sig-mul* (food), *meog-i* (feed), *yag* (drug)}, respectively (FIGURE 1). We connected the predicate to only the upper-level nodes that do not violate the exceptional cases below.

3- Handle exceptional cases in which the predicate cannot connect to an upper-level node.
   - If the upper-level node has children nodes that are not suitable with the predicate, connect the predicate with the argument noun itself. For instance, "*mul-jil*" is the upper-level node of the argument noun "*yag*." But, "*mul-jil*" also has children whose meaning cannot be eaten. So, we connected the predicate "*meog-da_020101*" directly with the argument "*yag*."
   - If a predicate has two or more argument nouns that are in hyponymy relations with each other, connect the predicate with the upper-level node of the hypernym. For instance, the argument noun "*eum-sig*" is a hypernym of the argument noun "*bab*" (FIGURE 1). Therefore, we connected "*meog-da_0201*" with "*eum-sig-mul*," which is the upper-level node of "*eum-sig*."
   - If argument nouns are in the same branch of a similar semantic field, connect the predicate with the common upper-level node. For instance, FIGURE 2 illustrates an upper-level node common to three argument nouns.
   - If the argument nouns are homographs, connect the predicate with upper-level nodes, depending on their semantics.
   - If the argument noun is a top-level node or its upper-level node is unsuitable for the predicate, we connect the predicate directly with the argument noun.

According to the principles, the subcategorization information was constructed with the number of predicates and LCS shown in the last column of Table 2.

C. CURRENT STATUS OF UWordMap

UWordMap now contains more than 474,000 words (nouns, verbs, adjectives, and adverbs), which is 92.2% of the words in SKLD. We compared UWordMap and existing Korean

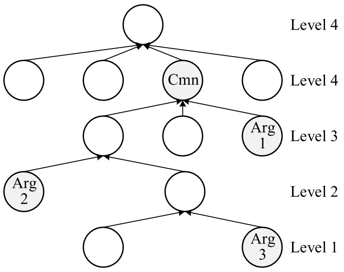

FIGURE 2. Example of common upper-level node. [Cmn] is the common upper-level node of the argument nouns [Arg1], [Arg2], [Arg3].

TABLE 2. Comparison of UWordMap and existing Korean word nets.

	KorLex	CorNet	LCN	SKLD	UWordMap
Noun	104,417	51,607	49,000	377,281	365,774 / 98,264
Verb	20,151	5,290	30,000	90,237	73,694 / 51,336
Adjective	20,897	2,801		21,618	16,853 / 12,438
Adverb	3,123			25,178	17,697 / 6,187
Total	150,199	58,985	79,000	514,314	474,018 / 168,225

The last column gives the total amount in UWordMap / amount in the subcategorization information.

word nets (KorLex 2.0 [31], CoreNet [26], and the lexical concept network (LCN) [27]). KorLex was translated from the English WordNet [20] by the Pusan National University. CoreNet was constructed based on the Japanese NTT Goidaikei [32] by the Korea Advanced Institute of Science and Technology. LCN was directly constructed for only nouns and verbs by the Electronics and Telecommunications Research Institute of Korea (ETRI). As shown in Table 2, UWordMap is the biggest and most comprehensive Korean LSN.

IV. KOREAN WORD SENSE DISAMBIGUATION

In this section, we propose a hybrid method for building a Korean WSD system that combines a corpus-based approach and a knowledge-based approach. First, we use the corpus-based approach to make statistics for the training corpus. However, the capacity of the training corpus is always limited, which causes the lack of data problem. Therefore, we also use the knowledge-based approach, specifically UWordMap, to expand the training corpus and improve WSD accuracy.

A. CORPUS-BASED WSD

The corpus-based WSD approach to Korean ordinarily involves two main stages [33]. The first stage generates candidates for each *eojeol*[3] using a morphological analysis and sense-code tagging. The second stage selects the most appropriate candidate based on its context.

In the first stage, we used the fast and accurate Corpus-based Korean Morphological Analysis (CKMA) method [34] to analyze the morphemes of each *eojeol*. CKMA first constructed a pre-analyzed partial *eojeol* dictionary based on the Sejong corpus to analyze *eojeols* and determine their morpheme compositions. Then, the morphemes were tagged with the part-of-speech (POS) using a hidden Markov model. CKMA was trained on 90% of the Sejong corpus and tested on the 10% remainder. The accuracy and recall of CKMA were 96.8% and 99.1%, respectively. The CKMA processed approximately 48,000 *eojeol(s)* per second on a CPU core i7 860 (2.8 GHz). Subsequently, we tagged the morphemes with all possible sense-codes from the SKLD to generate candidates.

In the second stage, the candidate selection for an *eojeol* is used to identify the correct sense for that *eojeol* (so-called WSD process). The examination of all *eojeols* in a sentence is infeasible because of low recall and high time consumption; therefore we examine only the two contiguous *eojeols* on the left and right and select a candidate that maximizes the conditional probability function:

$$WSD\,(w_i) = argmax_j\; P(c_{i,j}|w_{i-1}, w_i, w_{i+1}) \qquad (1)$$

where,

$$\begin{aligned} P(c_{i,j}|w_{i-1}, w_i, w_{i+1}) &\simeq P(c_{i,j}|w_{i-1}, w_i) \times P(c_{i,j}|w_i, w_{i+1}) \\ P_{Left} &= P(c_{i,j}|w_{i-1}, w_i) \\ P_{Right} &= P(c_{i,j}|w_i, w_{i+1}) \end{aligned}$$

So,

$$WSD\,(w_i) = argmax_j\; (P_{Left} \times P_{Right}) \qquad (2)$$

where w_i is the i-th *eojeol* (current *eojeol*) in a sentence $w_1 w_2 \ldots w_n$, and $c_{i,j}$ is the j-th candidate of the i-th *eojeol*. The probability $P(c_{i,j}|w_{i-1}, w_i)$ depends on the left-contiguous *eojeol* w_{i-1}, so it is denoted as P_{Left}. $P(c_{i,j}|w_i, w_{i+1})$ is calculated based on the right-contiguous *eojeol* w_{i+1} and is denoted as P_{Right}.

Table 3 gives an example of candidate generation for the *eojeol* "*sa-gwa-leul*," which appears in the sentence "*mas-iss-neun sa-gwa-leul meog-eoss-da*" (I ate a delicious apple). In this case, CKMA analyzed "*sa-gwa-leul*" into two morphemes "*sa-gwa*" and "*leul*" and POS-tagged "*sa-gwa*" as NNG (common noun) and "*leul*" as JKO (object marker). Subsequently, we looked up the probable sense-codes of the morphemes in SKLD and tagged two sense-codes "08" and "05" for "*sa-gwa*," which generated two candidates: $c_{2,1}$, "*sa-gwa_05*/NNG + *leul*/JKO" and $c_{2,2}$, "*sa-gwa_08*/NNG + *leul*/JKO." Furthermore, we assumed that the sense of the current *eojeol* can be identified based on the first two syllables of the contiguous *eojeols*. In the Korean writing system, an *eojeol* is a sequence of one or more syllables. Among them, the first syllables contain the meaning, and the last syllables often comprise one or more function words that reflect grammatical or structural relationships. For instance, in the phrases "*sa-gwa-leul meog-eoss-ji-man*" (I ate an apple, but) and "*sa-gwa-leul meog-eoss-da*" (I ate an apple), the function words "*ji-man*" (but) and "*da*" (sentence ending) are added

[3] *Eojeol* is the Korean token unit delimited by a white space. An *eojeol* consists a content word and one or more function words, such as postpositions, endings, auxiliaries, and predicates.

TABLE 3. An example of candidate generation.

w_1	w_2	w_3
mas-iss-neun (delicious)	*sa-gwa-leul* (apple)	*meog-eoss-da* (ate)
	$c_{2,1}$: *sa-gwa_05*/NNG + *leul*/JKO $c_{2,2}$: *sa-gwa_08*/NNG + *leul*/JKO	

$c_{2,1}$ and $c_{2,2}$ are candidates of the *eojeol* "*sa-gwa-leul*." /NNG and /JKO are tagged POS for a common noun and object marker, respectively. "*sa-gwa_05*" means an apple, whereas "*sa-gwa_08*" means an apology.

to the content word "*meog-eoss*" (ate). According to this assumption, the sense of "*sa-gwa*" can be identified based on the first two syllables of the following *eojeol* "*meog-eoss.*"

This assumption considers only the surface form of the *eojeols*. Hence, the conditional probability P_{Right} is denoted as P_{Right_Surf} and counted using Equation (3) based on the entire current *eojeol* and the first two syllables of the right-contiguous *eojeol*.

$$P_{Right\_Surf} = P(c_{i,j}|w_i, s_{i+1,1}, \quad s_{i+1,2}) \tag{3}$$

where $s_{x,y}$ is the y-th syllable of the x-th *eojeol*.

Likewise, the conditional probability P_{Left} is denoted as P_{Left_Surf} and calculated by Equation (4) based on the entire left-contiguous *eojeol* and the first two syllables of the current *eojeol*.

$$P_{Left\_Surf} = P\left(m_{i,j,1} \mid w_{i-1}, s_{i,1}, s_{i,2}\right)^U \times P(c_{i,j}|w_i) \tag{4}$$

where $m_{i,j,1}$ is the first morpheme (including the tagged POS and sense-code) in the j-th candidate of the i-th *eojeol*. For instance, in the example in Table 3, $m_{2,1,1}$ = "sa-gwa_05/NNG" and $M_{2,2,1}$ = "sa-gwa_08/NNG." U is a weight of P_{Left} to measure the relative importance of P_{Left} and P_{Right}. Because only the first two syllables of the current *eojeol* are involved in Equation (4), only the probability of the first morpheme in the current *eojeol* is calculated. The remaining morphemes are not considered. Therefore, we multiply the probability of the first morpheme by the probability of the current candidate given the entire current *eojeol*, $P(c_{i,j}|w_i)$, to make the probabilityP_{Left_Surf}.

Using the surface form of *eojeols* can improve the computational speed of these conditional probabilities, but it must deal with the data missing from the training corpus, which causes the low-recall problem in Korean WSD systems. In Korean sentences, the surface forms of verbs and adjectives are often constituted by adding function words to the word stems or transforming the original forms. Because of the many kinds of function words and many regular and irregular transformations, the training corpus cannot cover all the possible surface forms of each verb and adjective. For instance, when identifying the meaning of "*sa-gwa*" in the phrase "*sa-gwa-leul meog-ja-myeon,*" we cannot use Equation (4) because the pair of "*sa-gwa-leul*" and the first two syllables "*meog-ja*" do not exist in the training corpus. However, the pair of "*sa-gwa-leul*" and the verb "*meog_02/VV*" occurs many times in the training corpus. The verb "*meog_02/VV*" is the word stem for the *eojeol* "*meog-ja-myeon*." Based on the word stem of the verbs and adjectives, we can somewhat solve the missing data problem and determine the correct sense of the contiguous nouns.

A surface form of a verb or adjective can be analyzed into several kinds of word stems. For instance, the surface form "*gan-da*" is analyzed into four kinds of word stems, as shown in Table 4. In that case, P_{Right} is denoted as P_{Right_Stem} and calculated by selecting a word stem from the right-contiguous *eojeol* that maximizes the conditional probability:

$$P_{Right\_Stem} = argmax_k \; P(c_{i,j}|w_i, v_{i+1,k}) \tag{5}$$

where $v_{i+1,k}$ is the k-th word stem of the right-contiguous *eojeol* (i.e., i-th = current *eojeol*, i+1-th = right-contiguous *eojeol*). k can be mapped to 1, 2, 3, and 4 for the example in Table 4.

TABLE 4. Analyzing surface form of the ***Eojeol*** **"*gan-da*" into word stems.**

Word Stem	Function Word
v_1: *ga_01*/VV (to go)	*n-da*/EF
v_2: *ga_01*/VX (on going)	*n-da*/EF
v_3: *gal_01*/VV (to replace)	*n-da*/EF
v_4: *gal_02*/VV (to grind)	*n-da*/EF

v_1, v_2, v_3, and v_4 are word stems generated by analyzing the surface form of the *eojeol* "*gan-da*". /VV, /VX, and /EF are tagged POS referring to a verb, auxiliary verb, and final ending, respectively.

Likewise, when using word stems, the conditional probability P_{Left} is denoted as P_{Left_Stem} and calculated by Equation (6) by selecting a word stem for the current *eojeol* that maximizes the conditional probability:

$$P_{Left\_Stem} = argmax_k \; (P\left(m_{i,j,1} \mid w_{i-1}, v_{i,k}\right)^U \times P(c_{i,j}|w_i)) \tag{6}$$

This is equivalent to using the surface form in Equation (4), in which a word stem is always contained in the first morpheme of an *eojeol*. Therefore, only the first morpheme $m_{i,j,1}$ is calculated, and the other morphemes are not considered. We have to calculate the further probability of the candidate given entire current *eojeol, P*$(c_{i,j}|w_i)$.

In this research, we aim to build a fast and accurate WSD system, so we prioritize the surface form over the word stem of verbs and adjectives. We use the word stem to identify the word sense only if we fail to do that using the surface form. Generally, P_{Left} and P_{Right} are adjusted based on the use of surface forms or word stems as follows:

$$P_{Right} = \begin{cases} P_{Right\_Surf}, & if \; P_{Right\_Surf} > 0 \\ P_{Right\_Stem}, & if \; P_{Right\_Surf} = 0 \end{cases} \tag{7}$$

$$P_{Left} = \begin{cases} P_{Left\_Surf}, & if \; P_{Left\_Surf} > 0 \\ P_{Left\_Stem}, & if \; P_{Left\_Surf} = 0 \end{cases} \tag{8}$$

Even using the word stem of predicates (including verbs and adjectives), the corpus-based approach must still deal with the data missing from the training corpus. Each noun can combine with many different predicates to make up sentences. Given all the possible combinations of nouns and predicates, the training corpus cannot contain them all. That lack of data in the training corpus is one of the main challenges faced by the corpus-based approach to WSD.

B. KNOWLEDGE-BASED WSD

To address that problem, we propose a knowledge-based method using the UWordMap LSN. UWordMap contains a hierarchical structure network for nouns and subcategorization information for predicates. In the subcategorization information, predicates are connected with the LCS of the hierarchical noun network through sentence patterns. Table 5 gives an example of the subcategorization information in UWordMap. Based on those connections, we can determine the correct sense for nouns and predicates.

TABLE 5. Example of subcategorization information in UWordMap.

Predicate	Sentence Pattern	LCS (Arguments)
geod-da_02 (to walk)	*eul*	*gil_0101* (street), *geoli_0101* (avenue), *gong-won_03* (park), ...
geod-da_04 (to gather / to collect)	*e-ge-seo*	*baeg-seong_0001* (subjects, the people)
	e-seo	*si-heom-jang_0001* (exam place, test site), *jib_0101* (house), …
	eul	*seong-geum_03* (donation), *hoe-bi_03* (fee, dues), *ssal_0003* (rice), ...

Thus, UWordMap provides a way to complement the training corpus by generating sentences from the subcategorization information. From the subcategorization information, we extract the LCS (i.e., nouns), predicates, and sentence-patterns and then arrange them according to Korean sentence structure to generate sentences [35]. For instance, from the subcategorization information in Table 5, we can generate the following sentences to supplement the training corpus.

gil_0101/NNG *eul*/JKO *geod-da_02*/VV.
geoli_0101/NNG *eul*/JKO *geod-da_02*/VV.
gong-won_03/NNG *eul*/JKO *geod-da_02*/VV.
baeg-seong_0001/NNG *e-ge-seo*/SRC *geod-da_04*/VV.
si-heom-jang_0001/NNG *e-seo*/LOC *geod-da_04*/VV.
. . .

Furthermore, the training corpus can be expanded into hyponyms of the LCS on the LSN. The LCS's were replaced with their hyponyms to generate a series of sentences with the same predicates. FIGURE 3 gives a hierarchical network of the noun "gil_0101" (street, road). In the subcategorization information (Table 5), "*gil_0101*" is directly connected to the verb "*geod-da_02*" (to walk). However, its hyponyms (i.e., *mo-laes-gil_01*, *san-gil_02, mi-lo_0101*, and *na-mus-gil_01*. . . in FIGURE 3) are not connected to the verb "*geod-da_02*". We can still generate a series of sentences by connecting the verb "*geod-da_02*" and the hyponyms such as

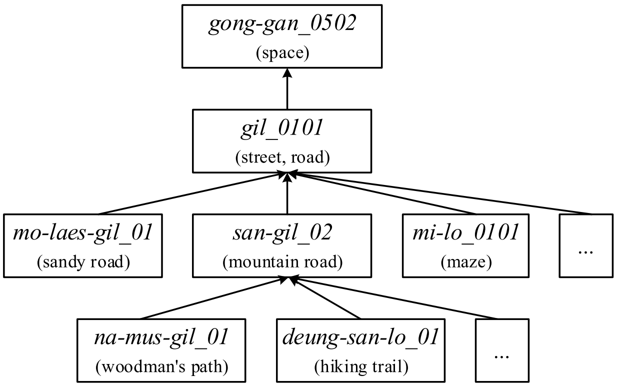

FIGURE 3. Hierarchical network of the noun "*gil_0101*" in UWordMap.

mo-laes-gil_01/NNG *eul*/JKO *geod-da_02*/VV.
san_gil_02/NNG *eul*/JKO *geod-da_02*/VV.
na-mus-gil_01/NNG *eul*/JKO *geod-da_02*/VV.
. . .

The noun "*gil_0101*" has 421 direct hyponyms (level 1 hyponyms), each of which has other hyponyms in level 2. Therefore, the number of sentences generated is large. Expanding the training corpus in this way will make it huge and cause low performance, reducing the system's practicality.

Instead of generating sentences to complement the training corpus, we replace the noun with its hypernym while the sentence is examining. When using both the surface form and the stem word of an *eojeol* fails to identify its sense, the hypernym will be looked up and used instead. If the hypernyms still cannot identify the sense of an *eojeol*, we continue looking up the hypernym of the hypernym in a looping process that continues until the sense is identified or the hypernym is the top-level node (FIGURE 4).

To improve the performance of the loop process, we make paths (hypernym paths) from each noun to a top-level node. Because each noun has only one hypernym, each noun has only one hypernym path. The average length of the hypernym paths is 10, and the maximum length is 17. For example, "*na-mus-gil_01* > *san-gil_02* > *gil_0101* > *gong-gan_0502*" is a hypernym path created from the hierarchical network shown in FIGURE 3. Storing the hypernym paths in the database could reduce the volume of the training corpus and reduce the complexity of looking up hypernyms in the loop process. All processes we have proposed for determining word sense are shown in FIGURE 4.

C. EXPERIMENTS AND RESULTS

We conducted our experiments on the Sejong corpus [36] and LSN UWordMap. The Sejong corpus includes 11 million *eojeols* tagged with POS and sense-codes that are identical to

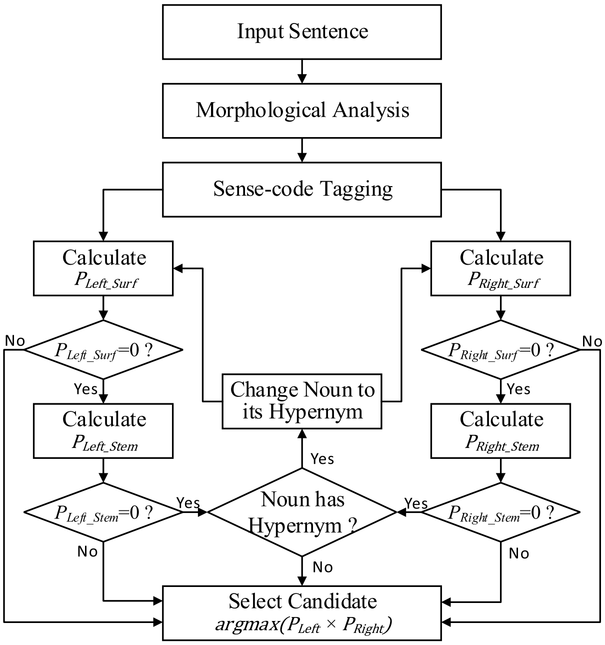

FIGURE 4. **WSD system architecture. Morphological analysis and sense-code tagging processes generate candidates. P_{Left_Surf} and P_{Left_Stem} are the probability of the candidates based on the surface form and the stem word of the left-contiguous *eojeol*, respectively. Likewise, P_{Right_Surf} and P_{Right_Stem} are the probability of candidates given the right-contiguous *eojeol*.**

TABLE 6. **Korean WSD results comparison.**

Approach	Accuracy	Time Consumption
PPWD	93.56%	23.0 sec
HMM	96.49%	44.1 sec
Proposed corpus-based	96.42%	33.0 sec
Proposed UWordMap association	**96.52%**	**37.0 sec**
Embedded Word Space	96.20%	N/A
Recurrent Neural Network	85.50%	N/A

The results were obtained when word sense disambiguating for 1,108,204 *eojeols* on the system of CPU core i7 860 (2.8 GHz).

those in SKLD. We also used the sense-codes from SKLD to construct UWordMap, so the sense-codes of UWordMap and the Sejong corpus are consistent.

To measure the weight U in Equations (4) and (6), we initialized U to 0.5 and increased its value to maximize the system accuracy. The accuracy increased as U increased from 0.5 to 1.5, and then it decreased as U increased from 1.5 to 2.5. Therefore, we used the weight $U = 1.5$ for all following experiments.

We used about 90% of the Sejong corpus for training and 10% for testing our systems. The testing dataset includes 1,108,204 *eojeols* extracted from the Sejong corpus by selecting sentences with orders divisible by 10. The rest of the corpus was used as the training dataset.

To evaluate both accuracy and performance, we set the same experiment environments and tested four systems using the following methods:

- PPWD: pre-analyzed partial word-phrase dictionary method [33].
- HMM: hidden Markov model method [37].
- Proposed corpus-based: our proposed method using only the corpus-based approach.
- Proposed associated UWordMap: our proposed method using a combination of the corpus-based approach and UWordMap.

The accuracy and time consumption of those systems in our testing using the dataset of 1,108,204 *eojeols* are shown in Table 6. The PPWD method consumed the least time but was not accurate enough to be a real system. Combining the corpus-based approach and UWordMap improved the accuracy by 0.1% compared with using only the corpus-based approach. Our proposed method significantly reduced the time consumed and achieved a higher accuracy compared with the HMM method.

We also compared the accuracy of our proposed method with the accuracies of recent machine learning methods: embedded word space (EWS) [38] and bidirectional recurrent neural network (BRNN) [39]. Both the EWS and BRNN methods used the Sejong corpus to train and evaluate their systems. The EWS method limited the training data to three POS: nouns, verbs, and adjectives. On the other hand, the BRNN method used all kinds of POS and extended the training data by adding corpora from Wikipedia and Namuwiki. As shown in Table 6, the proposed method outperformed both the EWS and BRNN methods.

V. NEURAL MACHINE TRANSLATION

NMT is the use of neural networks on parallel corpora to train a statistical model for machine translation that can regenerate a target sentence y by maximizing the conditional probability of y given a source sentence x. The use of neural networks to train translation models was first proposed in the 1990s [40], [41]. However, at that time, the hardware did not have enough power to handle the computational complexity, which caused the results to be unreasonable. Therefore, this method stalled for almost two decades.

Recently, with the development of hardware and deep learning technology, NMT has achieved state-of-the-art performance. Most NMT models are based on a sequence-to-sequence framework [4], [5] that uses two RNNs to encode a source sentence into a vector and then decode the vector into a target sentence. Attention mechanisms [7], [42], [43] were introduced to improve the translation results by dynamically customizing the RNN. For instance, Zhang *et al.* [44] altered the RNN to assemble history and future context into source sentences; Su *et al.* [45] segmented a source sentence into a word-clause-sentence hierarchical structure and then modified both the RNN encoder and decoder to input and translate the structure. The attention NMT architecture has now become the dominant paradigm.

Here, we describe the encoder–decoder from the attention NMT architecture proposed by Bahdanau *et al.* [7], upon which we built our MT system.

A. ENCODER

The encoder is a bi-directional RNN (i.e., forward and backward RNNs) that reads a source sentence into a sequence of context vectors. The source sentence is a sequence of 1-of-K coded word vectors $x = \left(x_1, x_2, \ldots, x_{T_x}\right)$, $x_i \in \mathbb{R}^{K_x}$, where T_x is the length of the source sentence, and K_x is the vocabulary of the source language.

The forward RNN reads the source sentence from left to right and computes forward hidden states $(\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_{T_x})$. The backward RNN reads the source sentence in the reverse order and produces backward hidden states $(\overleftarrow{h}_1, \overleftarrow{h}_2, \ldots, \overleftarrow{h}_{T_x})$.

The forward hidden state at time t is calculated by

$$\vec{h}_t = \begin{cases} (1 - \vec{z}_1) \circ \vec{h}_{\mathrm{t}-1} + \vec{z}_1 \circ \underline{\vec{h}}_t, & if\ t > 0 \\ 0, & if\ t = 0 \end{cases} \tag{9}$$

where

$$\underline{\vec{h}}_t = \tanh(\vec{W}\bar{E}x_t + \vec{U}[\vec{r}_t \circ \vec{h}_{\mathrm{t}-1}]) \tag{10}$$
$$\vec{z}_t = \sigma(\vec{W}_z\bar{E}x_t + \vec{U}_z\vec{h}_{\mathrm{t}-1}) \tag{11}$$
$$\vec{r}_t = \sigma(\vec{W}_r\bar{E}x_t + \vec{U}_r\vec{h}_{\mathrm{t}-1}) \tag{12}$$

$\bar{E}$ is a word-embedding matrix of the source language that is shared forward and backward, and $\vec{W}_*$ and $\vec{U}_*$ are weight matrices. σ denotes a logistic sigmoid function. The calculation of hidden backward states is similar to that for forward states.

The forward and backward hidden states are concatenated to have the source annotations $\left(h_1, h_2, \ldots, h_{T_x}\right)$ with

$$h_i = \left[\vec{h}_i^T ; \overleftarrow{h}_i^T\right]^T .$$

B. DECODER

A decoder is a forward RNN to generate the target sentence $\mathrm{y} = \left(y_1, y_2, \ldots, y_{T_y}\right), y_i \in \mathbb{R}^{K_y}$, where T_y is the length of target sentence, and K_y is the vocabulary of the target language. Word y_i is calculated by the conditional probability

$$p\left(y_i \mid \{y_1, \ldots, y_{i-1}\}, x\right) = g(y_{i-1}, s_i, c_i) \tag{13}$$

The hidden state is first initialized with $s_0 = \tanh(W_s h_1)$ and then calculated for each time i by

$$s_i = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i \tag{14}$$

where

$$\tilde{s}_i = tanh\left(WEy_{i-1} + U\left[r_i \circ s_{i-1}\right] + Cc_i\right) \tag{15}$$
$$z_i = \sigma\left(W_z Ey_{i-1} + U_z s_{i-1} + C_z c_i\right) \tag{16}$$
$$r_i = \sigma\left(W_r Ey_{i-1} + U_r s_{i-1} + C_r c_i\right) \tag{17}$$

E is the word-embedding matrix of the target language, and W_*, U_*, and C_* are weight matrices.

The context vector c_i is calculated based on the source annotations by

$$c_i = \sum\nolimits_{j=1}^{T_x} \frac{\exp\left(e_{ij}\right)}{\sum\nolimits_{k=1}^{T_x} \exp\left(e_{ik}\right)} h_j \tag{18}$$
$$e_{ij} = v_a^T \tanh(W_a s_{i-1} + U_a h_j) \tag{19}$$

where e_{ij} is an attention mechanism to measure how well h_j and y_i match, and v_a^T, W_a, and U_a are weight matrices.

VI. EXPERIMENTS AND RESULTS

To evaluate the effectiveness of our Korean WSD method in improving NMT results, we conducted a series of experiments using bi-directional translation between Korean and English, French, Spanish, and Japanese.

A. DATASETS

We built parallel corpora by extracting the definition statements of each word from the National Institute of Korean Language's Learner Dictionary.[4] After collecting the corpora, we normalized and preprocessed the sentences of each collected language. All sentences longer than 80 words were discarded. All alphabetical characters were lowercased for all languages. The corpora were basically tokenized using the Moses tokenizer[5] for Korean, English, French, and Spanish and the Mecab tokenizer[6] for Japanese.

In the end, we obtained 69,833 sentences for each language. These corpora are too small to build commercial MT systems, but they are large enough to allow us to evaluate the effectiveness of our Korean WSD method with NMT. We randomly extracted 1,000 sentences from each language to use as testing sets. The remainder of the corpora were used as training sets. Details of the corpora are shown in Table 7.

TABLE 7. Training and testing datasets.

	Training		Testing	
	#Tokens	#Types	#Tokens	#Types
Korean	497,463	45,272	8,652	3,840
English	886,777	18,403	13,194	3,228
French	919,820	22,128	14,125	2,947
Japanese	897,776	20,894	13,725	2,709
Spanish	712,731	28,296	14,680	2,682

B. INTEGRATING KOREAN WSD INTO THE CORPORA

By using the Korean WSD system described in section IV. The Korean words in both the training and testing sets were tagged with the sense-codes before they were input into the NMT systems. The Korean WSD system thus works as a preprocessor for MT systems. Table 8 gives an example of a Korean sentence tagged with the sense-codes. Because MT systems delimit words by the white spaces between them, the sense-code tagging transforms homographic words into distinct words, eliminating the ambiguous words from the Korean dataset.

The Korean WSD system changed the sizes of the tokens and vocabulary (i.e., the types of tokens) in the Korean dataset, as shown in Table 9. As explained in detail above, the Korean WSD includes two steps. The first step analyzes

[4] https://krdict.korean.go.kr
[5] http://www.statmt.org/moses
[6] http://taku910.github.io/mecab

TABLE 8. An example of a sense-code tagged sentence.

<table>
<tr><th>Original form</th><th>Sense-code tagged form</th></tr>
<tr><td>배를 먹고 배를 탔더니 배가 아팠다 .
(bae-leul meog-go bae-leul tass-deo-ni bae-ga a-pass-da .)</td><td>배_03|NNG 를|JKO 먹_02|VV 고|EC 배_02|NNG 를|JKO 타_02|VV 았|EP 더니|EC 배_01|NNG 가|JKS 아프|VA 았|EP 다|EF .|SF
(bae_03|NNG leul|JKO meog_02|VV go|EC bae_02|NNG leul|JKO ta_02|VV ass|EP deo-ni|EC bae_01|NNG ga|JKS a-peu|VA ass|EP da|EF .|SF)</td></tr>
</table>

TABLE 9. Korean data set after applying WSD.

	Training		Testing	
	#Tokens	#Types	#Tokens	#Types
Original	497,463	45,272	8,652	3,840
Morphological Analysis	895,261	12,914	15,834	2,146
WSD Tagging	895,261	14,035	15,834	2,247

the morphology into which a Korean word (*eojeol*) is segmented and then recovers it to the original form. The second step tags homographic words with the appropriate sense-codes. The morpheme segmentation increased the token size. The original form recovery reduced the vocabulary size. Tagging different sense-codes to the same homographic words increased the vocabulary size.

C. SETUP

We implemented our NMT systems on the open framework OpenNMT [46], which is a sequence-to-sequence model described in section V. The systems were set with the following parameters: word-embedding dimension = 500, hidden layer = 2x500 RNNs, input feed = 13 epochs.

We used those NMT systems for bi-directional translation of the following language pairs: Korean-English, Korean-French, Korean-Japanese, and Korean-Spanish. To separately evaluate the effectiveness of our morphological analysis and sense-code tagging, we used three systems (Baseline, Morphology, and WSD) for each direction. The Baseline systems were trained with the originally collected corpora given in Table 7. The Morphology systems were trained with the Korean corpus that had been morphologically analyzed. In the WSD systems, the Korean training corpus was both morphologically analyzed and tagged with sense-codes. Altogether, we had 24 translation systems, as shown in Table 10.

D. RESULTS

We used the BLEU, TER, and DLRATIO evaluation metrics to measure the translation quality. BLEU (Bi-Lingual Evaluation Understudy) [47] measures the precision of an MT system by comparing the n-grams of a candidate translation with those in the corresponding reference and counting the number of matches. In this research, we use the BLEU metric with 4-grams. TER (Translation Error Rate) [48] is an error metric for MT that measures the number of edits required to change a system output into one of the references. DLRATIO [49] (Damerau-Levenshtein edit distance) measures the edit distance between two sequences.

TABLE 10. Translation results.

Systems	BLEU	TER	DLRATIO
Korean-to-English Baseline	20.39	64.27	53.31
Korean-to-English Morphology	25.49	62.18	52.07
Korean-to-English WSD	**30.35**	**57.63**	**48.10**
English-to-Korean Baseline	23.49	71.03	58.39
English-to-Korean Morphology	24.05	68.58	56.18
English-to-Korean WSD	**27.48**	**62.86**	**52.28**
Korean-to-French Baseline	18.65	64.92	53.17
Korean-to-French Morphology	24.58	58.85	48.13
Korean-to-French WSD	**26.60**	**55.73**	**42.05**
French-to-Korean NMT	12.94	83.22	66.18
French-to-Korean Morphology	14.27	78.91	63.19
French-to-Korean WSD	**16.85**	**70.44**	**57.38**
Korean-to-Spanish Baseline	15.09	69.86	56.94
Korean-to-Spanish Morphology	20.56	63.82	52.40
Korean-to-Spanish WSD	**23.26**	**62.04**	**45.14**
Spanish-to-Korean Baseline	13.44	80.25	63.75
Spanish-to-Korean Morphology	14.41	77.87	62.42
Spanish-to-Korean WSD	**16.28**	**71.39**	**58.05**
Korean-to-Japanese Baseline	39.85	45.43	33.92
Korean-to-Japanese Morphology	48.98	34.49	26.94
Korean-to-Japanese WSD	**52.47**	**32.73**	**22.86**
Japanese-to-Korean Baseline	34.22	43.60	33.02
Japanese-to-Korean Morphology	42.76	38.47	29.24
Japanese-to-Korean WSD	**45.31**	**38.03**	**28.62**

Table 10 shows the results of the 24 systems in terms of their BLEU, TER, and DLRATIO scores. All three metrics demonstrate that both the Morphology and WSD systems improved the translation quality for all four language pairs and both translation directions.

The Morphology systems improved the results of the Baseline systems for all the language pairs by an average of 6.41 and 2.85 BLEU points for translation from and to Korean, respectively. Morphological complexity causes a critical data sparsity problem when translating into or from Korean [50]. The data sparsity increases the number of out-of-vocabulary words and reduces the probability of the occurrence of each word in the training corpus. For instance, NMT systems treat the morphologies of the Korean verb "*to go*" as completely different words: "*ga-da*," "*gan-da*," "*ga-yo*," and "*gab-ni-da*." Hence, the Korean morphological analysis can improve the translation results. The disproportionate improvement of results in different translation directions occurred because we applied the morphological analysis only to the Korean side. Therefore, the improvement of translations from Korean is more significant than that in the reverse direction.

The Korean sense-code tagging helped the NMT systems correctly align words in the parallel corpus as well as choose correct words for an input sentence. Therefore, the performance of the WSD systems further improved by an average of 3.27 and 2.61 BLEU points for all the language pairs when translating from and to Korean, respectively. In comparison with the Baseline systems, the WSD systems improved the translated results for all language pairs by an average of 9.68 and 5.46 BLEU points for translations from and to Korean, respectively. In summary, the proposed Korean WSD can remarkably improve the translation quality of NMT systems.

The TER and DLRATIO metrics provide more evidence that the proposed Korean WSD system can improve the translation quality of NMT. The results in Table 10 show that the proposed Korean WSD system improved the NMT performance by an average of 9.1 TER and 9.8 DLRATIO error points when translating from Korean to the four different languages. In the reverse direction, the proposed Korean WSD improved the performance by an average of 8.8 TER and 6.3 DLRATIO error points for all NMT systems. Particularly, the Korean sense-code tagging improved translation error prevention by 4.04 TER points and 4.51 DLRATIO points for all the language pairs. In short, the proposed Korean WSD can considerably reduce NMT errors.

Furthermore, we examined some well-known MT systems to see how they handle the Korean WSD problem. We input the sentence used in Section I, "*bae-leul meog-go bae-leul tass-deo-ni bae-ga a-pass-da*" into Google Translate, Microsoft Bing Translator, and Naver Papago. The translated results are shown in Table 11. Google Translate correctly translated the second and third "bae" but incorrectly translated the first "bae." Microsoft Translator could not distinguish the different meanings of "bae," and it missed a clause. Papago also could not distinguish the different meaning of "bae" in this sentence. None of them translated this sentence correctly.

TABLE 11. **Korean-to-English translation examples.**

Source	배를 먹고 배를 탔더니 배가 아팠다. (bae-leul meog-go bae-leul tass-deo-ni bae-ga a-pass-da .)
Original meaning	After eating a pear and getting on a boat, I had stomachache.
Papago	My stomach hurt after eating and riding on it.
Google Translate	When I ate a boat and got on the boat, my stomach hurt.
Microsoft Bing	He ate the boat, and the boat was sick.
Proposed system	I ate a pear and rode on the boat then stomach hurt.

Google Translate, Microsoft Bing, and Naver Papago were accessed on Apr. 25, 2018. Text highlighted in red is incorrectly translated.

VII. CONCLUSION

In this research, we have presented the following three accomplishments:

- We constructed the biggest and most comprehensive LSN for the Korean language — UWordMap, which is not only useful for MT, but also for various fields in Korean language processing, such as information retrieval and semantic webs.
- We proposed a method for building a fast and accurate Korean WSD system based on UWordMap.
- The experimental results from bi-directional translation between language pairs (Korean-English, Korean-French, Korean-Spanish, and Korean-Japanese) demonstrate that the proposed Korean WSD system significantly improved NMT results.

In the future, we plan to complete UWordMap with all the words contained in SKLD. We further intend to insert neologisms into UWordMap because adding more words will make the proposed Korean WSD system more accurate.

Because the quality of an NMT system depends on the training corpus, we also plan to collect more data related to Korean. Additionally, we intend to study the application of a syntactic and parsing attentional model to NMT system.

REFERENCES

[1] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico, "Neural versus phrase-based machine translation quality: A case study," in *Proc. EMNLP*, Austin, TX, USA, 2016, pp. 257–267.

[2] M. Junczys-Dowmunt, T. Dwojak, and H. Hoang, "Is neural machine translation ready for deployment? A case study on 30 translation directions," in *Proc. IWSLT*, Seattle, WA, USA, 2016, pp. 1–8.

[3] J. Crego *et al.* (Oct. 2016). "SYSTRAN's pure neural machine translation systems." [Online]. Available: https://arxiv.org/abs/1610.05540

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, Montreal, PQ, Canada, 2014, pp. 3104–3112.

[5] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1724–1734.

[6] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proc. EMNLP*, Seattle, WA, USA, 2013, pp. 1700–1709.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, San Diego, CA, USA, 2015.

[8] H. Choi, K. Cho, and Y. Bengio, "Context-dependent word representation for neural machine translation," *Comput. Speech Lang.*, vol. 45, pp. 149–160, Sep. 2017.

[9] R. Marvin and P. Koehn, "Exploring word sense disambiguation abilities of neural machine translation systems," in *Proc. AMTA*, Boston, MA, USA, 2018, pp. 125–131.

[10] M. Carpuat and D. Wu, "Word sense disambiguation vs. statistical machine translation," in *Proc. 43rd Annu. Meeting ACL*, Ann Arbor, MI, USA, 2005, pp. 387–394.

[11] D. Vickrey, L. Biewald, M. Teyssier, and D. Koller, "Word-sense disambiguation for machine translation," in *Proc. HLT/EMNLP*, Vancouver, BC, Canada, 2005, pp. 771–778.

[12] M. Carpuat and D. Wu, "Improving statistical machine translation using word sense disambiguation," in *Proc. EMNLP-CoNLL*, Prague, Czech Republic, 2007, pp. 61–72.

[13] D. Xiong and M. Zhang, "A sense-based translation model for statistical machine translation," in *Proc. 52nd Annu. Meeting ACL*, Baltimore, MD, USA, 2014, pp. 1459–1469.

[14] J. Su, D. Xiong, S. Huang, X. Han, and J. Yao, "Graph-based collective lexical selection for statistical machine translation," in *Proc. EMNLP*, Lisbon, Portugal, 2015, pp. 1238–1247.

[15] S. Neale, L. Gomes, E. Agirre, O. L. de Lacalle, and A. Branco, "Word sense-aware machine translation: Including senses as contextual features for improved translation models," in *Proc. LREC*, Ljubljana, Slovenia, 2016, pp. 2777–2783.

[16] R. Sudarikov, O. Dušek, M. Holub, O. Bojar, and V. Kríž, "Verb sense disambiguation in machine translation," in *Proc. COLING*, Osaka, Japan, 2016, pp. 42–50.

[17] Š. Vintar and D. Fišer, "Using WordNet-based word sense disambiguation to improve MT performance," in *Hybrid Approaches to Machine Translation*. Springer, 2016, pp. 191–205.

[18] A. Rios, L. Mascarell, and R. Sennrich, "Improving word sense disambiguation in neural machine translation with sense embeddings," in *Proc. WMT*, Copenhagen, Denmark, 2017, pp. 11–19.

[19] F. Liu, H. Lu, and G. Neubig. (Mar. 2018). "Handling homographs in neural machine translation." [Online]. Available: https://arxiv.org/abs/1708.06510

[20] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[21] P. Vossen, "Introduction to EuroWordNet," *Comp. Humanities*, vol. 32, nos. 2–3, pp. 73–89, Mar. 1998.

[22] D. Tufis, D. Cristea, and S. Stamou, "BalkaNet: Aims, methods, results and perspectives. A general overview," *Romanian J. Inf. Sci. Technol.*, vol. 7, nos. 1–2, pp. 9–43, 2004.

[23] Z. Dong and Q. Dong, *HowNet and the Computation of Meaning*. River Edge, NJ, USA: World Scientific, 2006.

[24] A. S. Yoon *et al.*, "Construction of Korean WordNet," *J. KIISE, Softw. Appl.*, vol. 36, no. 1, pp. 92–126, 2009.

[25] C. K. Lee and G. B. Lee, "Using WordNet for the automatic construction of Korean thesaurus," in *Proc. 11th Annu. Conf. Hum. Lang. Techn.*, 1999, pp. 156–163.

[26] K.-S. Choi, "CoreNet: Chinese-Japanese-Korean WordNet with shared semantic hierarchy," in *Proc. NLP-KE*, Beijing, China, 2003, pp. 767–770.

[27] M. Choi, J. Hur, and M.-G. Jang, "Constructing Korean lexical concept network for encyclopedia question-answering system," in *Proc. ECON*, Busan, South Korea, 2004, pp. 3115–3119.

[28] Y. J. Bae and C. Y. Ock, "Introduction to the Korean word map (UWordMap) and API," in *Proc. 26th Annu. Conf. Human Lang. Technol.*, 2014, pp. 27–31.

[29] D. A. Cruse, *Lexical semantics*. Cambridge, U.K.: Cambridge Univ. Press, 1986.

[30] M. J. Kim, "Does Korean have adjectives?" *MIT Work. Papers Linguistics*, vol. 43, pp. 71–89, 2002.

[31] A. S. Yoon, "Korean WordNet, KorLex 2.0—A language resource for semantic processing and knowledge engineering," in *Proc. HAN-GEUL*, vol. 295, 2012, pp. 163–201.

[32] S. Ikehara *et al.*, "The semantic system, volume 1 of Goi–Taikei—A Japanese Lexicon," Iwanami Shoten, Tokyo, Japan, Tech. Rep., 1997.

[33] J. C. Shin and C. Y. Ock, "Korean homograph tagging model based on sub-word conditional probability," *KIPS Trans. Softw. Data Eng.*, vol. 3, no. 10, pp. 407–420, 2014.

[34] J. C. Shin and C. Y. Ock, "A Korean morphological analyzer using a pre-analyzed partial word-phrase dictionary," *KIISE, Softw. Appl.*, vol. 39, no. 5, pp. 415–424, 2012.

[35] J. C. Shin and C. Y. Ock, "Improvement of Korean homograph disambiguation using Korean lexical semantic network (UWordMap)," *J. KIISE*, vol. 43, no. 1, pp. 71–79, 2016.

[36] H. Kim, "Korean national corpus in the 21st century Sejong project," in *Proc. NIJL*, Tokyo, Japan, 2006, pp. 49–54.

[37] J. C. Shin and C. Y. Ock, "A stage transition model for Korean part-of-speech and homograph tagging," *KIISE, Softw. Appl.*, vol. 39, no. 11, pp. 889–901, 2012.

[38] M. Y. Kang, B. Kim, and J. S. Lee, "Word sense disambiguation using embedded word space," *J. Comput. Sci. Eng.*, vol. 11, no. 1, pp. 32–38, Mar. 2017.

[39] J. H. Min, J. W. Jeon, K. H. Song, and Y. S. Kim, "A study on word sense disambiguation using bidirectional recurrent neural network for Korean language," *J. Korea Soc. Comput. Inf.*, vol. 22, no. 4, pp. 41–49, 2017.

[40] M. A. Castano, F. Casacuberta, and E. Vidal, "Machine translation using neural networks and finite-state models," in *Proc. 7th Inter. Conf. TMI*, 1997, pp. 160–167.

[41] M. L. Forcada and R. P. Ñeco, "Recursive hetero-associative memories for translation," in *Proc. Inter. Work-Conf. Artif. Neural Netw.*, Berlin, Germany, 1997, pp. 453–462.

[42] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, Lisbon, Portugal, 2015, pp. 1412–1421.

[43] S. Jean, O. Firat, K. Cho, R. Memisevic, and Y. Bengio, "Montreal neural machine translation systems for WMT'15," in *Proc. 10th Workshop SMT*, Lisbon, Portugal, 2015, pp. 134–140.

[44] B. Zhang, D. Xiong, J. Su, and H. Duan "A context-aware recurrent encoder for neural machine translation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 12, pp. 2424–2432, Dec. 2017.

[45] J. Su *et al.*, "A hierarchy-to-sequence attentional neural machine translation model," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 3, pp. 623–632, Mar. 2018.

[46] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. (2017). "OpenNMT: Open-source toolkit for neural machine translation." [Online]. Available: https://arxiv.org/abs/1701.02810

[47] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. ACL*, 2002, pp. 311–318.

[48] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proc. Assoc. MT Americas*, Boston, MA, USA, 2006, pp. 223–231.

[49] G. V. Bard, "Spelling-error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string-edit distance metric," in *Proc. ACSW*, Ballarat, VIC, Australia, 2007, pp. 117–124.

[50] Q. P. Nguyen, J. C. Shin, and C. Y. Ock, "Korean morphological analysis for Korean-Vietnamese statistical machine translation," *Elect. Sci. Technol.*, vol. 15, no. 4, pp. 413–419, 2017.

QUANG-PHUOC NGUYEN received the B.S. degree in information technology from the University of Natural Sciences, part of Vietnam National University, Ho Chi Minh City, Vietnam, in 2005, and the M.S. degree in information technology from Konkuk University, Seoul, South Korea, in 2010. He is currently pursuing the Ph.D. degree with the University of Ulsan, Ulsan, South Korea. His research interests include natural language processing, machine learning, and machine translation.

ANH-DUNG VO received the B.S. degree in computer science from the Hanoi University of Science and Technology, Vietnam, in 2010, and the M.S. degree in information technology from the University of Ulsan, Ulsan, South Korea, in 2013, where he is currently pursuing the Ph.D. degree. His research interests include natural language processing, machine learning, and sentiment analysis.

JOON-CHOUL SHIN received the B.S., M.Sc., and Ph.D. degrees in information technology from the University of Ulsan, Ulsan, South Korea, in 2007, 2009, and 2014, respectively. He is currently a Post-Doctoral Researcher with the University of Ulsan. His research interests include Korean language processing, document clustering, and software engineering.

CHEOL-YOUNG OCK received the B.S., M.S., and Ph.D. degrees in computer engineering from the National University of Seoul, South Korea, in 1982, 1984, and 1993, respectively, and the Honorary Doctorate degree from the School of IT, National University of Mongolia, in 2007. He has been a Visiting Professor with the Russia Tomsk Institute, Russia, in 1994, and Glasgow University, U.K., in 1996. He was a Chairman of sigHCLT (2007–2008) in KIISE, South Korea. He has been a Visiting Researcher with the National Institute of Korean Language, South Korea, since 2008. He is currently a Professor with the School of IT Convergence, University of Ulsan, South Korea. He has been constructing the Ulsan Word Map since 2002. His research interests include natural language processing, machine learning, and text mining. He received the Medal for Korean development from the Korean Government in 2016.