Word Sense Induction with Attentive Context Clustering

Moshe Stekel
Computer Science Dpt.,
Ariel University,
Israel
mstekel@gmail.com

Amos Azaria
Computer Science Dpt.,
Ariel University,
Israel
amos.azaria
@ariel.ac.il

Shai Gordin
Land of Israel Studies
and Archaeology Dpt.,
Ariel University, Israel
shygordin@gmail.com

Abstract

In this paper we present ACCWSI (Attentive Context Clustering WSI), a method for Word Sense Induction, suitable for languages with limited resources. Pretrained on a small corpus and given an ambiguous word (query word) and a set of excerpts that contain it, ACCWSI uses an attention mechanism for generating context-aware embeddings, distinguishing between the different senses assigned to the query word. These embeddings are then clustered to provide groups of main common uses of the query word. We show that ACCWSI performs well on the SemEval-2 2010 WSI task. ACCWSI also demonstrates practical applicability for shedding light on the meanings of ambiguous words in ancient languages, such as Classical Hebrew and Akkadian.

1 Introduction

Natural language expresses human concepts, thoughts, emotions and insights. That is, natural language represents a model of extremely high complexity—the human mind (at least, its communication-driven layers). Some researchers believe that natural language is an environment in which compromise is inevitable when projecting the infinite number of dimensions of human thinking onto the much smaller number of dimensions of human speech (Fedorenko and Varley, 2016). Multiplicity of meaning of a single word, such as polysemy (similarity obtained from a common source) or homonyny (accidental similarity), is therefore an expected product of this compromise. Below are two common examples of word sense ambiguity:

- "I can hear *bass* sounds" versus "They like grilled *bass*"
- "We crossed the river to the other *bank*" versus "Mike deposited the money in his *bank* account"

Humans are able to disambiguate the polysemy/homonyny or understand contextual nuances by using clues that come from the context of the ambiguous word. One of the fundamental tasks of natural language processing is Word Sense Induction (WSI), a task of automatic discrimination of different senses of words by finding these contextual clues.

It is difficult to overestimate the importance of accurate Word Sense Induction when dealing with common Natural Language Processing (NLP) tasks, such as Information Retrieval or Search Clustering. Furthermore, historical research seeks to correctly induce the meaning of words in order to resolve doubts about many historical issues. As a good example we can refer to the Akkadian lemma "galû", the meaning of which ranges between the negative shade of "exile" or "deportation", the neutral shade of "relocation" and the positive one of "appointment". Another example is the Hebrew lemma "zakar", which takes on both the meanings of "memory" and "male". Accurate Word Sense Induction is essential for correct understanding of ancient documents.

In this paper, we present an Attentive Context Clustering WSI (ACCWSI). ACCWSI first creates a word-embedding for each word, which is identical for any context that it appears in. ACCWSI uses the cosine similarity between the words in the context and the word in focus to determine the attention that each word should achieve to form a context aware vector representation for each appearance of the word in focus. ACCWSI then clusters the resulting vectors, such that each cluster represents a different meaning of the word. ACCWSI has demonstrated high practical applicability in languages with limited resources and obtained a very high score by the evaluation framework of SemEval-2 2010 Task 14 (Manandhar et al., 2010). ACCWSI achieved a high score not only with the

Proceedings of the Workshop on Natural Language Processing for Digital Humanities (NLP4DH), pages 144–151
co-located with ICON-2021, December 16-19, 2021, Silchar, India.
https://doi.org/10.26615/978-952-94-5833-2_017

original training dataset, but also with a training dataset reduced to a fraction of 2.6% of the original dataset, which is comparable to the size of the Hebrew Bible.

2 Related Work

Word Sense Induction and Word Sense Disambiguation provide fertile ground for researchers, starting from very early attempts to tackle these non-trivial tasks, such as "simulated annealing" according to human-edited dictionary (Cowie et al., 1992) and employing the "conceptual distance" between contexts (Agirre and Rigau, 1996), going through later unsupervised methods, that use patterns of word co-occurrence (Bordag, 2006) or bigrams of web search results (Udani et al., 2005), continuing with "hidden concepts" of the contextual words, that not necessarily overlap with the sense of the ambiguous word (Chang et al., 2014), and ending with the most recent solutions like (Eyal et al., 2021), that uses word substitutions of modern Masked Language Models, such as Google BERT MLM.

Our research was inspired by two main works: the context-group discrimination algorithm (Schütze, 1998) from the Context Clustering category and the Google BERT language model (Vaswani et al., 2017). Amrami and Goldberg (Amrami and Goldberg, 2019) utilize Google BERT for their WSI method. However, their method does not meet our requirement of being able to induce word senses in languages with limited resources, as training Google BERT on small corpora does not provide sufficient accuracy (Ezen-Can, 2020). The high scores achieved by the BertWSI model in the SemEval-2 2010 Task 14 (Manandhar et al., 2010) metrics are credited to the fact that the underlying model was pre-trained by Google on a huge corpus of text. Our solution takes advantage of the basic mechanism of attention (Galassi et al., 2020) underlying BERT without applying the complex process of learning attention weights and thus achieves good results when applied to small datasets. The only weight learning process we use is the Word2Vec (Goldberg and Levy, 2014) model training that requires far fewer resources than attention-based learning. Thus, we provide a practical tool in the study of the meanings of words in resource-limited languages, such as ancient dead languages. The Clustering by Committee work (Pantel and Lin, 2002) gave us the idea to use a threshold of 0.5 as an acceptable proportion of orphan instances when measuring the quality of a clustering solution (see Section 3.4.3). We also explored Lin's algorithm (Lin, 1998), which uses the word clustering approach by combining words with similar semantics into sense representations, but it was found less effective when it came to discriminating senses of words in resource-constrained languages.

3 Task and Algorithm

3.1 WSI task definition

The general definition of WSI is automatic detection of the set of senses denoted by a word. A simplified version of WSI can be defined as follows: given a list of lemmatized sentences and a query lemma, find all the sentences in the list that contain the query lemma, and group them so that the instances of the query lemma in one group are semantically similar to each other and noticeably different from the instances in other groups. This is a simplified definition because, when lemmatizing, we ignore some input information, such as the part of speech, tense etc. Note that ignoring the part of speech information of the target word is attractive, especially for ancient genres in which the archaic syntactic forms of words may provide no part of speech information (for instance refer to some hardly explainable verses of the Hebrew Psalms).

3.2 Attention mechanism

Our method uses the following "basic attention" mechanism: given a target word (query) and its "context", either the whole sentence or some "window" of words containing the query word, each element of the context is evaluated by its cosine similarity to the query word. The result is optionally multiplied by a constant factor and eventually softmaxed. We refer to the result as the "weights of similarity" or "weights of relevance". The closer two words are semantically, the greater is the cosine similarity between their embeddings and, therefore, the appropriate weights of relevance are greater. The original word embeddings of the context members are multiplied by the appropriate weights of relevance and thus the power of every context member is improved or worsened according to its relevance to the query word. When these new context-sensitive embeddings are summed into a single vector, this sum represents a context-aware vector of

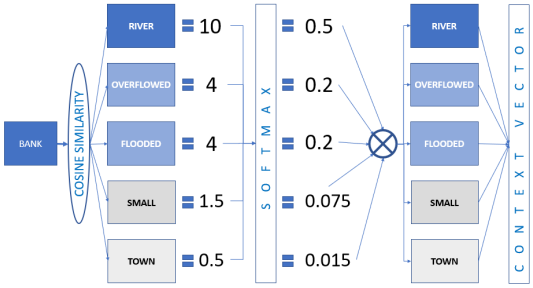

Figure 1: Illustration of the attention mechanism

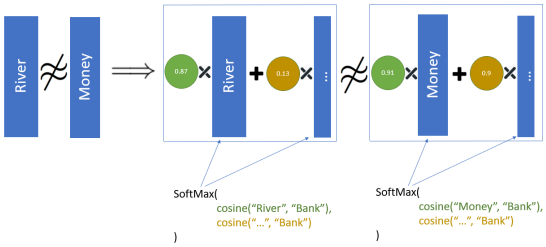

Figure 2: An illustration of separability of context-aware vectors generated by ACCWSI: the most relevant terms (green weights) with respect to the query term "bank" are "river" in the first context and "money" in the second context. They are different and therefore the result context-aware vectors are different. Less relevant terms are multiplied by smaller weights (light brown) and thus have smaller effect on the final context-aware vector.

the query word that embeds its "local sense" with respect to this specific context, where the relevance of each context member is taken into consideration. Figure 1 illustrates this mechanism.

3.3 The ACCWSI algorithm

We now present our Attentive Context Clustering WSI (ACCWSI) algorithm. The ACCWSI algorithm (see Algorithm 1) first replaces the lemmas with their Word2Vec embeddings (Goldberg and Levy, 2014). It then uses the attention mechanism described above (Section 3.2), resulting in context-aware vectors, that are used as input to the DBSCAN clustering algorithm (Schubert et al., 2017), producing clusters of different "shades of meaning" of the query lemma. Since different contexts are best defined by different most relevant context members, and conversely - similar contexts are defined by similar context members, the result vectors can be easily clustered. Figure 2 illustrates this idea.

3.4 Hyperparameters

Algorithm 1 uses several hyperparameters: Word2Vec window, the choice of the clustering algorithm and the internal hyperparameters of the

Algorithm 1 ACCWSI

Input:
text ▷ a list of lemmatized sentences
lemma ▷ a query lemma
Output:
context groups of the query lemma

$model \leftarrow word2vec(text)$
$sentences \leftarrow filter\_by\_lemma(text, lemma)$
$ctx\_aware\_vecs \leftarrow []$
for each $s \in sentences$ **do**
  $ctx\_vecs \leftarrow model.get\_vectors(s)$
  $lemma\_vec \leftarrow model.get\_single\_vector(lemma)$
  $sim \leftarrow cosine\_sim(ctx\_vecs, lemma\_vec)$
  $sim\_weights \leftarrow softmax(sim)$
  $new\_lemma\_vec \leftarrow \sum_i ctx\_vecs_i * sim\_weights_i$
  $ctx\_aware\_vecs.push(new\_lemma\_vec)$
end for
return $DBSCAN(...).fit(ctx\_aware\_vecs)$

latter. The optimal values of these parameters can be found either empirically or by using well-known optimization methods. In this section we explain these hyperparameters, briefly overview the optimization methods, and present the method that achieved best accuracy in our case.

3.4.1 Word2Vec Window

This parameter determines the size of the context to be scanned from each direction around the target word when training the Word2Vec model to perform the missing word prediction task (CBoW architecture) or the context prediction task (Skip-Gram architecture). The optimal value of this parameter intuitively depends on the native average "density of context" inherent to the target language. We found the optimal value empirically by iterating over the range from 2 to 10 and evaluating the result by manually checking the semantic similarity of words suggested by the model. The best values were 5 for English and 2 for Classical Hebrew. This difference is probably due to the specific syntactic structures of Classical Hebrew verses, which are statistically much shorter than the syntactic structures of typical Modern English sentences.

3.4.2 The choice of the clustering algorithm

We evaluated several different clustering algorithms on our task, including KMeans (Hamerly and Elkan, 2004), Gaussian-Mixture model (Reynolds, 2009) and DBSCAN (Schubert et al., 2017). DBSCAN performed slightly better and was therefore selected as our clustering algorithm.

3.4.3 DBSCAN-eps

This parameter is a key one for the density-based clustering proposed by DBSCAN. It defines the maximum distance between two points to be considered as neighbors. There are several methods in the literature for optimizing the value of this parameter, such as the Kneedle algorithm for finding the maximum curvature in the graph of distances, the Silhouette Score for evaluating the clustering quality, and more. Although these optimization methods demonstrated good performance (unsupervised V-Measure of 15.3%), we propose a heuristic that performed better. The rationale behind the heuristic is that text can contain instances of ambiguous words with highly clear context, in addition to other instances with more obscure context. Decreasing the value of eps results in clearer but tighter clusters, filtering out distant "noisy" instances. In our case, narrowing the clusters while keeping the number of the "noisy" instances below 50% gave good results. Algorithm 2 demonstrates this heuristic.

Algorithm 2 Fine-tuning the DBSCAN eps hyperparameter - the value of eps is iteratively decreased until the noise (the fraction of the orphan instances) becomes greater than $\frac{1}{2}$

$best\_eps \leftarrow 0.95$
for each $x \in range(90, 0, -5)$ **do**
  $eps \leftarrow x/100$
  $labels \leftarrow DBSCAN(eps = eps)$
  $.fit(cxt\_aware\_vectors)$
  $noise \leftarrow labels.count(-1)/len(labels)$
  if $noise \leq 0.5$ **then**
    $best\_eps \leftarrow eps$
  else
    break
  end if
end for
return $best\_eps$

4 Experimental evaluation

We ran an experiment to evaluate the algorithm on **Sem-Eval 2010 Task 14** (Manandhar et al., 2010), which aims to objectively measure and compare the quality of WSI systems. Both training and test data are English sentences containing polysemous or homonymous nouns and verbs. The goal of the task is to split the instances of each ambiguous word and their contexts into clusters representing different meanings. The result is assessed by comparison with the "Gold Standard" clustering performed by human experts. In Section 4.1 we present the Unsupervised V-Measure and F-Score metrics of this assessment as well as the Supervised Recall metric.

4.1 SemEval-2 2010 Task 14 Evaluation

In Task 14 of the SemEval-2 2010 workshop (Manandhar et al., 2010), participants were asked to train their models on the corpus of training data provided by the organizers, and then perform word sense induction for a set of sentences containing both ambiguous nouns and ambiguous verbs. The results were assessed against the "Gold Standard" clusters compiled by human experts. The tables below show the metrics achieved with ACCWSI trained on the full training corpus (by training ACCWSI we mean training its internal Word2Vec model), as well as the metrics achieved with the reduced ACCWSI, which was trained on a randomly selected 2.6% of the training data, along with those of the participants with the highest scores in every metric.

System	VM % (All)	VM % (Nouns)	VM % (Verbs)
ACCWSI full	17.3	20.7	12.3
Hermit	16.2	16.7	15.6
UoY	15.7	20.6	8.5
KSU KDD	15.7	18	12.4
ACCWSI reduced	15.4	18.8	10.4
Duluth-WSI	9	11.4	5.7
...			
...			
Duluth-WSI-SVD-Gap	0	0	0.1

Table 1: V-Measure (VM) unsupervised evaluation. V-Measure assesses the quality of a clustering solution by explicitly measuring its homogeneity and its completeness. Homogeneity refers to the degree that each cluster consists of data points primarily belonging to a single Gold Standard class, while completeness refers to the degree that each Gold Standard class consists of data points primarily assigned to a single cluster. V-Measure is the harmonic mean of the homogeneity and completeness.

System	FS % (All)	FS % (Nouns)	FS % (Verbs)
Duluth-WSI-SVD-Gap	63.3	57	72.4
KCDC-PT	61.8	57	72.4
...			
...			
ACCWSI reduced	55.9	51.3	62.7
...			
...			
ACCWSI full	53.8	47.2	63.4
Duluth-WSI-SVD	41.1	37.1	48.2
Duluth-WSI	41.1	37.1	48.2
...			
...			
Duluth-R-110	16.1	15.8	16.4

Table 2: Paired F-Score (FS) unsupervised evaluation: two sets of instance pairs are generated - a set of all possible instance pairs within each induced cluster and a set of all possible instance pairs within each Gold Standard class. Precision is the number of common instance pairs between the two sets to the total number of pairs in the induced clusters, while recall is the number of common instance pairs between the two sets to the total number of pairs in the Gold Standard classes. F-Score is the harmonic mean between precision and recall.

System	SR % (All)	SR % (Nouns)	SR % (Verbs)
ACCWSI full	63.7	59.6	71.1
ACCWSI reduced	62.7	57.5	69.8
UoY	62.4	59.4	66.8
Duluth-WSI	60.5	54.7	68.9
...			
...			
Duluth-Mix-Uni-Gap	18.7	1.6	43.8

Table 3: Supervised recall (SR) using a test set split with 80% mapping and 20% evaluation. In this evaluation, the testing dataset is split into a mapping and an evaluation corpus. The first one is used to map the automatically induced clusters to Gold Standard senses, while the second one is used to evaluate methods in a WSD setting.

5 Application examples

In this section we present examples of applying our method to a relatively small Hebrew corpus—the Hebrew Bible. We used the text-fabric version of the BHSA project to generate the appropriate dataset and run the ACCWSI algorithm on it. Figure 3 shows the operation of the ACCWSI algorithm used to obtain two different meanings of "bank" in English. Figure 4 and Figure 5 present the induced classes for two ambiguous Hebrew Biblical lemmas: **khalal** (dead body/desecrate) and **zakar** (male/memory). The instances of the first lemma were split into 2 sense clusters while the instances of the second lemma were split into 5 sense clusters. ACCWSI seems to perform well and provide satisfactory clusters despite the small training corpus.

6 Future work

Iterating the process of generating context embeddings may improve the accuracy of the clustering. In our future work we plan to develop a method for determining the "center of mass" (or "centroid" for convex clusters) of every cluster. These centers will be treated as new "query" embeddings and the ACCWSI attention-weighted technique will be reapplied within each cluster using its new query (its center). This should provide finer discrimination of meanings. This iterative process can be repeated many times until maximum accuracy is achieved.

Another effort we lead these days is Word Sense Induction in ancient Akkadian texts. Between the 9th to the late 7th centuries BCE, the Assyrian Empire deported millions of people across the Near East. By even the most humble estimates, around 1.3 million people were moved around as a result of conquest, labour recruitment or as punishment, just to name the central reasons for this dire process (Sano 2020). However, the records for these deportations are numerous and came down to us in different genres that deal with the act of deportation, or forced migration, from different points of view: contemporaneous Assyrian royal inscriptions, letters and administrative texts, as well as Babylonian historical chronicles, written many years after the events in question. All were written in Assyrian and Babylonian, two close dialects of Akkadian, the oldest known (East-)Semitic language in the world. In all, 19 different verbs deal with various stages of the forced migration, like the capture of people or forced recruitment, their change of location, and resettlement. Even then, there are differences across meanings for specific verbs, sometimes minute ones, but also quite substantial in terms of semantics.

A good example of such a complicated verb is *galû* which the Chicago Assyrian Dictionary (CAD), the most comprehensive dictionary of Akkadian, translates as "1. to go into exile, 2. to deport, to exile (Š-stem, causative)" (CAD Š/3, 201). Its usage is limited to a Babylonian context, either in Assyrian letters dealing with Babylonia or Babylonian chronicles (Sano 2020, 34). As text 1 shows, the usage, much like that of Biblical Hebrew *GLY/H*, is used in consequence of a military

Sentence	Attention Highlights	Cluster
Her bank account was rarely over two hundred.	account, rarely	0
After breakfast, she closed her account at the bank and turned in her resignation.	account, close, turn	0
How could a man with four million in the bank be in financial danger?	financial, man, danger	0
Seating herself on a low bank, she studied the souls.	seat, study, low	0
If you would know the history of these homesteads, inquire at the bank where they are mortgaged.	mortgage, homestead, history	0
I guess he had some bucks at one time – back when he bought all this land – but his bank account never held a candle to mine.	account, hold, buy	0
A stream bank is the terrain alongside the bed of a stream	stream, stream, bed	1
He walked up and down the river, leading his hourse behind him; but he kept his eyes turned always toward the dim, dark spot which he knew was the old North Church.	river. church, spot	1
She waded to the bank and picked up he shoes and stockings.	stocking, shoe, wade	1
The town of Barwani is situated near the left bank of the Nerbudda	town, near, left	1
Cushing himself swam to the swamps on the river bank, and after wading among them for hours reached a Federal picket boat.	river, boat, swamp	1
Within an hour, there were riding side-by-side down the south bank of the creek, searching for the blocked area.	creek, area, south	1

Figure 3: Two different meanings of **bank**, the financial institute and the geographic terrain, are represented by the clusters in the figure. The "attention highlight" column shows the most relevant context words. The first cluster contains an interesting failure: the fourth sentence is clustered as a financial institute even though a human would cluster it as a geographic terrain. The reason is that the most relevant context words "seat, study, low" are not sufficiently indicative

.

Sentence	Attention Highlights	Cluster
וּמִזַּרְעֲךָ לֹא־תִתֵּן לְהַעֲבִיר לַמֹּלֶךְ וְלֹא תְחַלֵּל אֶת־שֵׁם אֱלֹהֶיךָ אֲנִי יְ־הוָ־ה.	נתן, י־ה־ו־ה, עבר	0
לָכֵן אֱמֹר לְבֵית יִשְׂרָאֵל כֹּה אָמַר אֲדֹנָי יְ־הוִ־ה לֹא לְמַעַנְכֶם אֲנִי עֹשֶׂה בֵּית יִשְׂרָאֵל כִּי אִם לְשֵׁם קָדְשִׁי אֲשֶׁר חִלַּלְתֶּם בַּגּוֹיִם אֲשֶׁר בָּאתֶם שָׁם.	גוי, ישראל, קדש	0
וְאֶחְמֹל עַל שֵׁם קָדְשִׁי אֲשֶׁר חִלְּלוּהוּ בֵּית יִשְׂרָאֵל בַּגּוֹיִם אֲשֶׁר בָּאוּ שָׁמָּה.	חמל, גוי, ישראל	0
וּבוֹא אֶל הַגּוֹיִם אֲשֶׁר בָּאוּ שָׁם וְיַחֲללוּ אֶת שֵׁם קָדְשִׁי בֶּאֱמֹר לָהֶם עַם יְ־הוָ־ה אֵלֶּה וּמֵאַרְצוֹ יָצָאוּ.	עם, גוי, קדש	0
וְנָפְלוּ חֲלָלִים בְּאֶרֶץ כְּשִׂדִּים וּמְדֻקָּרִים בְּחוּצוֹתֶיהָ.	דקר, חוץ, נפל	1
כִּי נָתַתִּי אֶת חִתִּיתִי בְּאֶרֶץ חַיִּים וְהֻשְׁכַּב בְּתוֹךְ עֲרֵלִים אֶת חַלְלֵי חֶרֶב פַּרְעֹה וְכָל הֲמוֹנָה נְאֻם אֲדֹנָי יְ־הוִ־ה.	ערל, חרב, נאם	1
אוֹתָם יִרְאֶה פַּרְעֹה וְנִחַם עַל כָּל הֲמוֹנוֹ חַלְלֵי חֶרֶב פַּרְעֹה וְכָל חֵילוֹ נְאֻם אֲדֹנָי יְהוִה.	חרב, נאם, כל	1
שַׂמָּה נְסִיכֵי צָפוֹן כֻּלָּם וְכָל צִדֹנִי אֲשֶׁר יָרַד אֶת חֲלָלִים בַּחִתִּיתָם מִגְבוּרָתָם בּוֹשִׁים וַיִּשְׁכְּבוּ עֲרֵלִים אֶת חַלְלֵי חֶרֶב וַיִּשְׂאוּ כְלִמָּתָם אֶת יוֹרְדֵי בוֹר.	ערל, כל, כל	1

Figure 4: In the Hebrew Bible, the lemma **khalal** normally takes on the sense of either **dead body**(as a noun) or **desecrate**(as a verb). This figure presents the appropriate clusters generated by ACCWSI. The "attention highlight" column shows the most relevant context words. In the context of **desecrate** (cluster 0), the attention is paid to words like **God**, **sacred**, **nation** etc. while in the context of **dead body** (cluster 1), the attention is paid to **sword**, **stab**, **fall** etc.

conflict. However, a single instance in a letter from the time of Tiglath-pileser III (c. 731-730 BCE), here text 2, shows that under certain political circumstances people could ask for someone to deport them to Assyria, perhaps referring to the safety of being a protected refugee under the direct responsibility of the Assyrian king. This might also be the meaning of certain cases in Aramaic, where *gly* in G-stem active participle means "exile, refugee", or in D-stem means "to emigrate", (Comprehensive Aramaic Lexicon, s.v. gly D and C).

Text 1: SAA 19, 27 rev. 4'-8a' (online edition, Luukko 2012) 4' LUGAL? *lu*? *ú-di* NIM.MA.KI_-a_-[*a*] 5' LÚ.ERIM-MEŠ-*šú-nu* TA DUMU ^{m}GIN—NUMUN *la*? 6' *i-du-ku ù ša—da-a-ni* 7' *ú-sag-li-šú-nu šú-nu-ú-ma* 8' *ig-da-al-ú*

Sentence	Attention Highlights	Cluster
וּבֶן שְׁמֹנֶת יָמִים יִמּוֹל לָכֶם כָּל זָכָר לְדֹרֹתֵיכֶם יְלִיד בַּיִת וּמִקְנַת כֶּסֶף מִכֹּל בֶּן נֵכָר אֲשֶׁר לֹא מִזַּרְעֲךָ הוּא.	דור, יום, כל	0
פְּקֻדֵיהֶם בְּמִסְפַּר כָּל זָכָר מִבֶּן חֹדֶשׁ וָמָעְלָה פְּקֻדֵיהֶם שִׁבְעַת אֲלָפִים וְחָמֵשׁ מֵאוֹת.	מספר, שבע, פקד	0
בְּמִסְפַּר כָּל זָכָר מִבֶּן חֹדֶשׁ וָמָעְלָה שִׁמְנַת אֲלָפִים וְשֵׁשׁ מֵאוֹת שֹׁמְרֵי מִשְׁמֶרֶת הַקֹּדֶשׁ.	מספר, מעל, כל	0
וּפְקֻדֵיהֶם בְּמִסְפַּר כָּל זָכָר מִבֶּן חֹדֶשׁ וָמָעְלָה שֵׁשֶׁת אֲלָפִים וּמֵאתָיִם.	מספר, פקד, מעל	0
זְכָר רַחֲמֶיךָ יְ-ה-ו-ה וַחֲסָדֶיךָ כִּי מֵעוֹלָם הֵמָּה.	עולם, חסד, רחמים	1
חַטֹּאות נְעוּרַי וּפְשָׁעַי אַל תִּזְכֹּר כְּחַסְדְּךָ זְכָר לִי אַתָּה לְמַעַן טּוּבְךָ יְ-ה-ו-ה.	פשע, חסד, נעורים	1
פְּנֵי יְהוָה בְּעֹשֵׂי רָע לְהַכְרִית מֵאֶרֶץ זִכְרָם.	כרת, ה', רע	1
אֱלֹהַי עָלַי נַפְשִׁי תִּשְׁתּוֹחָח עַל כֵּן אֶזְכָּרְךָ מֵאֶרֶץ יַרְדֵּן וְהַרְמֹנִים מֵהַר מִצְעָר.	אלהים, נפש, כן	1
וְאִם מִן הַצֹּאן קָרְבָּנוֹ לְזֶבַח שְׁלָמִים לַי-ה-ו-ה זָכָר אוֹ נְקֵבָה תָּמִים יַקְרִיבֶנּוּ.	נקבה, תמים, ה'	2
אוֹ הוֹדַע אֵלָיו חַטָּאתוֹ אֲשֶׁר חָטָא בָּהּ וְהֵבִיא אֶת קָרְבָּנוֹ שְׂעִיר עִזִּים זָכָר תָּמִים.	חטא, תמים, קרבן	2
לְהַעֲלֹת תָּמִים זָכָר בַּבָּקָר בַּחֻשִּׁים וּבַעֲדָיִם.	תמים, רצון, כשב	2
וְאִם זֶבַח שְׁלָמִים קָרְבָּנוֹ אִם מִן הַבָּקָר הוּא מַקְרִיב אִם זָכָר אִם נְקֵבָה תָּמִים יַקְרִיבֶנּוּ לִפְנֵי יְהוָה.	נקבה, תמים, ה'	2
וַיָּבֹא אֶלְיָקִים בֶּן חִלְקִיָּהוּ אֲשֶׁר עַל הַבַּיִת וְשֶׁבְנָא הַסֹּפֵר וְיוֹאָח בֶּן אָסָף הַמַּזְכִּיר אֶל חִזְקִיָּהוּ קְרוּעֵי בְגָדִים וַיַּגִּידוּ לוֹ אֶת דִּבְרֵי רַב שָׁקֵה.	ספר, בן, אסף	3
וַיֵּצֵא אֶלְיָקִים בֶּן חִלְקִיָּהוּ אֲשֶׁר עַל הַבַּיִת וְשֶׁבְנָא הַסֹּפֵר וְיוֹאָח בֶּן אָסָף הַמַּזְכִּיר.	ספר, בן, אסף	3
אֲלִיתֹרֶף וַאֲחִיָּה בְּנֵי שִׁישָׁא סֹפְרִים יְהוֹשָׁפָט בֶּן אֲחִילוּד הַמַּזְכִּיר.	ספר, בן, יהושפט	3
וַיִּקְרָאוּ אֶל הַמֶּלֶךְ וַיֵּצֵא אֲלֵהֶם אֶלְיָקִים בֶּן חִלְקִיָּהוּ אֲשֶׁר עַל הַבַּיִת וְשֶׁבְנָא הַסֹּפֵר וְיוֹאָח בֶּן אָסָף	ספר, בן, אסף	3
וְהַיָּמִים הָאֵלֶּה נִזְכָּרִים וְנַעֲשִׂים בְּכָל דּוֹר וָדוֹר מִשְׁפָּחָה וּמִשְׁפָּחָה מְדִינָה וּמְדִינָה וְעִיר וָעִיר וִימֵי הַפּוּרִים הָאֵלֶּה לֹא יַעַבְרוּ מִתּוֹךְ הַיְּהוּדִים וְזִכְרָם לֹא יָסוּף מִזַּרְעָם.	דור, דור, יום	4
יְ-ה-ו-ה שִׁמְךָ לְעוֹלָם יְ-ה-ו-ה זִכְרְךָ לְדֹר וָדֹר	דור, דור, עולם	4
אַזְכִּירָה שִׁמְךָ בְּכָל דֹּר וָדֹר עַל כֵּן עַמִּים יְהוֹדוּךָ לְעוֹלָם וָעֶד.	דור, דור, שם	4
וְאַתָּה יְהוָה לְעוֹלָם תֵּשֵׁב וְזִכְרְךָ לְדֹר וָדֹר.	דור, דור, עולם	4

Figure 5: In the Hebrew Bible, the senses of the lemma **zakar** are related to either **male** or **memory**. This figure presents the five clusters generated by ACCWSI. The "attention highlight" column shows the most relevant context words. The first cluster represents the sense of **male human**, the second one - **God's memory**, the third one - **male animal sacrifice**, the fourth - **the role of scribe** and the fifth - **chronological memory**

(rev. 4'-5') The Elamites killed their soldiers with the son of Mukin-zeri and (6'-8') **deported them by force**. They too **went into exile**.

Text 2: SAA 19, 87 obv. 8b'-13a' (online edition, Luukko 2012) 8' *. . . e-gir-tum ša ina* UGU 9' [md]AMAR.UTU—A—SUM-*na na-u-ni-ni it-tab-lu-ni* 10' [*ina*] *pa-ni-ni i-si-si-ú* : *ù* m*ba-la-su* 11' [*ip*]-*ta-la-a a—da-niš ma-a an-nu-rig* x+[x x] 12' [*at*]-*tu-nu tal-la-ka ma-a ša-ga-la-ni* [o] 13' [*i*]-*si-ku-nu la-al-lik* . . .

(obv. 8'-9') They intercepted the letter which was brought to Merodach-baladan (10') and read it [in] our [pr]esence. But Balassu (11') [g]ot very scared, saying: (12') "You (pl.) must come this moment and **deport me!** (13') I will go [wit]h you (pl.)."

7 Conclusion

In this paper we propose ACCWSI, an algorithm to automatically induce various senses of ambiguous words by automatically focusing on the most relevant words from their contexts. After learning generic word embeddings into a Word2Vec model, ACCWSI uses the basic attention technique for determining the most relevant context members and generating context-aware embeddings, each with a semantic direction that aggregates the directions of its context members. Distant meanings imply distant context embeddings and vice versa, and thus standard clustering techniques can be easily applied for grouping the context embeddings by their common semantic directions. ACCWSI has shown excellent performance even when trained on a small subset of the training data in the SemEval-2 2010 task 14. Furtheremore, ACCWSI demonstrated high applicability in disambiguation of word senses in ancient Semitic languages, such as Classical Hebrew and Akkadian.

Acklowledgments

This research is supported by the Ministry of Science & Technology, Israel, Grant 3-16464.

References

Eneko Agirre and German Rigau. 1996. Word sense disambiguation using conceptual density. *arXiv preprint cmp-lg/9606007*.

Asaf Amrami and Yoav Goldberg. 2019. Towards better substitution-based word sense induction. *arXiv preprint arXiv:1905.12598*.

Stefan Bordag. 2006. Word sense induction: Triplet-based clustering and automatic evaluation. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Baobao Chang, Wenzhe Pei, and Miaohong Chen. 2014. Inducing word sense with automatically learned hidden concepts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 355–364.

Jim Cowie, Joe Guthrie, and Louise Guthrie. 1992. Lexical disambiguation using simulated annealing. In *COLING 1992 Volume 1: The 14th International Conference on Computational Linguistics*.

Matan Eyal, Shoval Sadde, Hillel Taub-Tabib, and Yoav Goldberg. 2021. Large scale substitution-based word sense induction. *arXiv preprint arXiv:2110.07681*.

Aysu Ezen-Can. 2020. A comparison of lstm and bert for small corpus. *arXiv preprint arXiv:2009.05451*.

Evelina Fedorenko and Rosemary Varley. 2016. Language and thought are not the same thing: evidence from neuroimaging and neurological patients. *Annals of the New York Academy of Sciences*, 1369(1):132.

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2020. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Greg Hamerly and Charles Elkan. 2004. Learning the k in k-means. *Advances in neural information processing systems*, 16:281–288.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 768–774.

Suresh Manandhar, Ioannis Klapaftis, Dmitriy Dligach, and Sameer Pradhan. 2010. Semeval-2010 task 14: Word sense induction &disambiguation. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 63–68.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619.

Douglas A Reynolds. 2009. Gaussian mixture models. *Encyclopedia of biometrics*, 741:659–663.

Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. 2017. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123.

Goldee Udani, Shachi Dave, Anthony Davis, and Tim Sibley. 2005. Noun sense induction using web search results. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 657–658.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.