Masrad: Arabic Terminology Management Corpora with Semi-Automatic Construction

Mahdi Nasser[1], **Laura Sayah**[1], **Fadi A. Zaraket**[1,2]
[1]Arab Center for Research and Policy Studies, Doha, Qatar
[2]American University of Beirut, Beirut, Lebanon
{mnasser, lsayyah, fzaraket}@dohainstitute.edu.qa

Abstract

This paper presents Masrad (i.e. "glossary" in Arabic), a terminology dataset for Arabic terminology management, and a method with supporting tools for its semi-automatic construction. The entries in Masrad are (f,a) pairs of foreign (non-Arabic) terms f, appearing in specialized, academic and field-specific books next to their Arabic a counterparts. Masrad-Ex systematically extracts these pairs as a first step to construct Masrad. Masrad helps improve *term consistency* in academic translations and specialized Arabic documents, and automate cross-lingual text processing. Masrad-Ex leverages translated terms organically occurring in Arabic books, and considers several candidate pairs for each term phrase. The candidate Arabic terms occur next to the foreign terms, and vary in length. Masrad-Ex computes lexicographic, phonetic, morphological, and semantic similarity metrics for each candidate pair, and uses heuristic, machine learning, and post-processing approaches to decide the best candidate. We make Masrad available to the interested research community. The best performing Masrad-Ex approach achieved 90.5% precision and 92.4% recall.

Keywords: Terminology management, Cross-lingual text processing, Terminology extraction, Arabic NLP

1. Introduction

Terminology management concerns (i) specifying terms used upon the introduction of novel concepts to a language, and (ii) enforcing the use, hence (iii) ensuring *consistency* and correctness. Translation from foreign languages into Arabic causes significant terminology inconsistency due to the diversity of translation experts and the lack of reference parallel term corpora. Recent systems like TURJUMAN (Nagoudi et al., 2022) have improved Arabic machine translation quality, but do not focus on terminology alignment or consistency. Linguistic and structural complexities affect consistency and clarity of translations and may hinder access to knowledge. The first and most important step is to build a termbase and provide it as a reference to Arabic scholars and experts.

We present Masrad, a large dataset of parallel terms with pairs of foreign terms and their Arabic counterparts, sourced into use in professional, academic and scholastic Arabic contexts. We make Masrad available for the research community and authors looking for consistent Arabic terms when introducing novel concepts to Arabic.

Recent advances in natural language processing (NLP) and computational linguistics (CL) have paved the way for automated systems that assist in cross-lingual text processing. However, existing systems, including LLMs, often struggle with the nuances of terminology in cross-lingual specific expert fields, where precision is paramount. This paper presents a novel methodology and a supporting framework (Masrad-Ex) to extract and curate parallel foreign-Arabic terms from existing Arabic books, and leverages to construct Masrad.

When a foreign term occurs in Arabic scholarly writing, it usually occurs between parentheses, and it is usually preceded by its Arabic translation. Two problems arise. (i) The exact scope of the Arabic term is ambiguous, and (ii) often times authors (even the same author) use Arabic terms inconsistently for the same non-Arabic ones. Editors and reviewers, who are usually expert researchers themselves, are required to manually solve these laborious and time consuming tasks during lengthy editing iterations.

Masrad and Masrad-Ex help automate these two tasks to reduce review and edit efforts and help reviewers and editors concentrate on more important research and knowledge generation tasks.

The problem is formulated as follows. Consider a sequence $s = \langle (f) w_1 w_2 w_3 \dots w_n \rangle$ representing a sentence with a foreign term f between parentheses, preceded from right to left by a sequence of Arabic words $w_1, \dots w_n$. We denote the candidate terms possible to match f as the set of strings $S(f) = \{s_i : s_i = w_1 \dots w_i, i \in [1,n]\}$ ending before the term f. Masrad-Ex extracts these candidates from the books and identifies the target candidate term $s_t \in S(f)$ that best matches f. Masrad-Ex extracts linguistic features for each candidate match (s_i, f). It interpolates the features and aggregates their values into a resulting score, and then ranks the candidate terms.

This work addresses the challenge of terminology mapping and reduces language barriers, facilitating a greater exchange of knowledge and ideas between languages and cultures.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 7918–7926
11–16 May 2026.

We make Masrad available for researchers upon request. We make the constructed dataset of feature vectors (used for training/testing) available online for the research community [1].

To the best of our knowledge, this approach to termbase building for Arabic is novel, and no comparable datasets currently exist.

1.1. Motivating example

Consider the following two contexts. They contain the same underlined french term **l'ethnocentrisme** (The ethnocentrism) with different translations.

- Context 1:
  ولا يتحول نقده الحاد للمركزية ـ الإثنية (l'ethnocentrisme) هنا إلى نسبية ثقافية، بل إلى كونية عقلية...
  (wlA ytHwl nqdh AlHAd llmrkzyp - Al<vnyp (l'ethnocentrisme) hnA <lY nsbyp vqAfyp, bl <lY kwnyp Eqlyp...[2] - "His sharp criticism of ethnocentrism (l'ethnocentrisme) does not turn into cultural relativism...")
- Context 2:
  انتقد كثيراً النزعة الإثنية المركزية (l'ethnocentrisme) التي تريد «إحلال أفكارنا الأوروبية محل أفكار الإنسان البدائي عن العالم والمجتمع»، وتحذّر من خطر «نسبة الأفكار المتقدمة إلى الإنسان البدائي لأنها تتجاوز إمكاناته العقلية».
  (Antqd kvyrAF AlnzEp Al<vnyp Almrkzyp (l'ethnocentrisme) Alty tryd '<HlAl >fkArnA Al>wrwbyp mHl >fkAr Al<nsAn AlbdA}y En AlEAlm wAlmjtmE', wtH*ř mn xTr 'nsbp Al>fkAr Almtqdmp <lY Al<nsAn AlbdA}y l>nhA ttjAwz <mkAnAth AlEqlyp'. - "He strongly criticized the ethnocentric tendency (l'ethnocentrisme) that seeks to replace primitive human ideas about the world and society with our European ideas. ")

Masrad-Ex computers the following candidates per context. **Candidates for Context 1:**

- ولا يتحول نقده الحاد للمركزية ـ الإثنية (wlA ytHwl nqdh AlHAd llmrkzyp - Al<vnyp - "And his sharp criticism of ethnocentrism does not turn into")
- يتحول نقده الحاد للمركزية ـ الإثنية (ytHwl nqdh AlHAd llmrkzyp - Al<vnyp - "His sharp criticism of ethnocentrism turns into")
- نقده الحاد للمركزية ـ الإثنية (nqdh AlHAd llmrkzyp - Al<vnyp - "His sharp criticism of ethnocentrism")
- الحاد للمركزية ـ الإثنية (AlHAd llmrkzyp - Al<vnyp - "The sharp ethnocentrism")
- المركزية ـ الإثنية (target term) (Almrkzyp - Al<vnyp - "Ethnocentrism")
- الإثنية (Al<vnyp - "Ethnicity")

Candidates for Context 2:

- انتقد كثيراً النزعة الإثنية المركزية (Antqd kvyrAF AlnzEp Al<vnyp Almrkzyp - "He strongly criticized the ethnocentric tendency")
- كثيراً النزعة الإثنية المركزية (kvyrAF AlnzEp Al<vnyp Almrkzyp - "The strong ethnocentric tendency")
- النزعة الإثنية المركزية (AlnzEp Al<vnyp Almrkzyp - "The ethnocentric tendency")
- الإثنية المركزية (target term) (Al<vnyp Almrkzyp - "Ethnocentrism")
- المركزية (Almrkzyp - "Centralism")

The two contexts and the extracted translation candidate terms showcase the inconsistency across contexts, and also expose potential adoption of different candidates by readers of each context, thus amplifying the problem.

This paper makes the following contributions:

- We propose a novel method for terminology mapping that leverages naturally occurring term translations, combining NLP and data driven techniques.
- We build and provide MARSAD, a unique annotated dataset for terminology extraction, which can serve as a valuable resource for the research community.
- We build and evaluate Masrad-Ex including three approaches: heuristic, machine learning (ML), and ML with post-processing demonstrating the effectiveness of these methods in addressing terminology mapping challenges.

The results are satisfying to experts in the Arab Center for Research and Policy Studies (ACRPS) and ready for initial deployment. The final model runs locally, requires no LLM interactions, and thus requires no data exposure or intellectual property risks.

We discuss related work in Section 2, Masrad in Section 3, the methodology in Section 4, and

[1] https://acr.ps/1L9Bakw

[2] Buckwalter encoding is used throughout the paper.

Arabic books	495
Unique foreign terms	58,570
Foreign term occurrences	84,242
Total candidates	334,564
Candidates per occurrence	3.97

Table 1: Statistics of all processed books

Arabic books	15
Unique foreign terms	4,347
All occurrences of foreign terms	4,841
Total candidates	19,405

Table 2: Statistics of MASRAD (annotated candidates)

present and discuss results in Section 5. We then discuss future directions and conclude.

2. Related work

TURJUMAN (Nagoudi et al., 2022) is a neural machine translation toolkit that translates into Modern Standard Arabic using fine-tuned AraT5 models. While effective at sentence-level translation, it does not address terminology alignment or consistency in specialized texts.

In addition to bilingual dictionaries, efforts have been made to develop lexical sources from several original languages into Arabic. Arabic WordNet (Global WordNet Association) for instance relies on several open and restricted domain sources to develop a lexical English-Arabic database in different fields. Other more specialized glossaries have been developed, such as Hossam Mahdi's Glossary of Terms for the Conservation of Cultural Heritage in Arabic Alphabetical Order (Mahdy). Furthermore, glossaries offering support to interpreters (Sharif) have also been established.

The Arabic digital ontology (Jarrar, 2022) includes digitized Arabic dictionaries as denoted by dictionary authors. It requires API access for term lookup.

However, these are not complete, need to be updated manually regularly for novel terms, and are not easy to integrate in the editing process.

Previous research in automatic term extraction frequently relies on parallel corpora (Dagan and Church, 1994; Aker et al., 2013). While effective for high-resource languages, such approaches are limited by the availability of strictly aligned texts. To address this constraint, our work explores the extraction of organically occurring parenthetical translations. Instead of requiring aligned cross-lingual document pairs, we identify translations that naturally co-occur within a single text stream when authors provide in-line clarifications.

3. Dataset construction

The source dataset is derived from books published by ACRPS centers in Doha, and Beirut. These books span multiple social sciences and humanities academic fields, offering a diverse range of terminologies. The dataset includes terms in foreign languages (enclosed in parentheses) alongside their surrounding Arabic text. The language of the foreign term doesn't matter for our purposes (e.g it can be English, French, German, etc.).

3.1. Source data selection

The texts we selected ensured diversity and relevance as they spanned multiple domains including History, Education, Religion, Politics, Sociology and Anthropology, Philosophy, Arts and Literature, Law, Language, Memoirs and Biographies.

We identified and extracted candidate terms to be all strings of words, of a specified length proportional relative to the foreign term, immediately preceding a parenthetical foreign term. This ensures that we almost always captured all potential translations for further analysis.

3.2. Annotated dataset construction

Each extracted candidate term, along with its associated foreign term, was presented as an individual instance. An **instance of the dataset** is the features of a candidate, and a label that is *True* if the candidate is the target candidate and *False* otherwise. The features of this instance, extracted using NLP tools, are described in Section 4.

Table 1 summarizes the data extracted from all books. Table 2 summarizes the data in Masrad, our labeled and expert reviewed data that will be used later for training Masrad-Ex.

To facilitate the labeling process, a draft of the labels was generated using heuristics (Section 5.1). This draft was then reviewed to correct errors.

4. Methodology

Figure 1 illustrates the flow of our methodology. First, the tool extracts the candidates from a context (a paragraph) with a foreign term. Then, the feature vector of each candidate is computed. Finally, each vector is fed to a model which outputs a score that helps in choosing the target translation of the source foreign term.

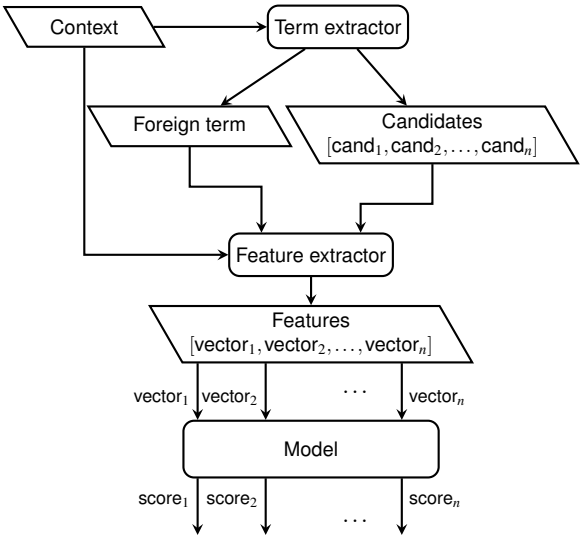

Figure 1: An overview of the process

Feature	Value
Semantic similarity	[0, 1]
Translation lexical similarity	[0, 1]
Transliteration lexical similarity	[0, 1]
Entity	{PER, LOC, ORG, MISC}
POS	{adj, adv, conj, misc, noun, noun_prop, part, prep, pron, verb}
Phonetic similarity	{False, True}

Table 3: Features extracted for each candidate

4.1. Feature extraction

Given a candidate s_i, its foreign term f, and its context c, we extract features of the candidate that will be used later to predict which candidate is the target term. These features with their possible values are listed in Table 3 and will be explained in the following subsections.

4.1.1. Semantic similarity

We utilized the sentence transformer (Reimers and Gurevych, 2019) LaBSE (Feng et al., 2022) to compute the embeddings of the source term and candidate term.

Candidate	Semantic similarity
آن هيرزبرغ (|n hyrzbrg - “Anne Herzberg”)	0.635
هيرزبرغ (hyrzbrg - “Herzberg”)	0.66

Table 4: Semantic similarity example

After that, we compute semantic similarity as the cosine similarity between these 2 vectors (Salton et al., 1975). This might seem sufficient but it's not, which is due to the high similarity between candidates (given any two candidates, one of them is a substring of the other). Therefore, we need more features to handle peculiar cases.

Table 4 shows how the semantic similarity alone score favors the 2^{nd} candidate, which missed the first name in the foreign term and is incorrect according to experts.

4.1.2. Translation lexical similarity

Translation lexical similarity uses automated translation of the foreign term and measures a lexical distance between that and the candidate terms. The translation tool used is GoogleTranslator from the deep-translator python package (Baccouri). The distance of choice is the Levenshtein (Levenshtein, 1966) distance which computes how many changes are needed for the terms to be identical. We use a ratio with respect to length to normalize the distance across candidates.

Example: suppose the foreign term is “London School of Economics and Political Science". Automatic translation returns
$t =$ كلية لندن للاقتصاد والعلوم السياسية
(klyp lndn llAqtSAd wAlElwm AlsyAsyp - “London School of Economics and Political Science.”)

We have the candidate:
$s =$ كلية لندن للعلوم الاقتصادية والسياسية
(klyp lndn llElwm AlAqtSAdyp wAlsyAsyp - “London School of Economics and Political Science.”)
The Levenshtein distance between s and t is 15. The ratio is calculated as

$$\frac{\text{len}(t) + \text{len}(s) - 15}{\text{len}(t) + \text{len}(s)} = 0.79$$

4.1.3. Transliteration lexical similarity

This feature captures the cases where the target Arabic term is just a transliteration of the source term by computing the Levenshtein ratio between the transliteration of the source term and the candidate. This is helpful in the case of translating proper nouns that should supposedly map phonetically. The transliteration tool used is the google-transliteration-api python package (GokulNC and Prem27).

Example: suppose the foreign term is "Ehud Prawer". After transliterating it, we get
$t =$ ايهود براور
(Ayhwd brAwr - “Ehud Brower”)
We also have the candidate with a hamza instead of an alef

$s =$ إيهود براور
(<yhwd brAwr - “Ehud Brower”)
The Levenshtein distance between s and t is 1. The ratio is calculated as

$$\frac{\text{len}(t) + \text{len}(s) - 1}{\text{len}(t) + \text{len}(s)} = 0.95$$

4.1.4. Entity

A named entity is a real-world object or concept that can be clearly identified and categorized with a proper name.

Candidates that are detected as named entities have a higher probability of being a term. We use an open source Arabic NER model (Jarrar et al., 2022) to detect entities in Arabic context. Also, NER is used on the source term to favor candidates having the same entity type; we used the spaCy (Honnibal et al., 2020) model xx_ent_wiki_sm in particular.

4.1.5. Part of speech

A part of speech (POS) is a category of words with similar grammatical properties. Common parts of speech include nouns, verbs, adjectives, adverbs, pronouns, prepositions, conjunctions, and interjections. Terms are mostly noun phrases and rarely start with verbs.

We used POS tagging tools (Obeid et al., 2020) with its maximum likelihood expectation morphological disambiguator to detect the POS of the first word occurring in a candidate.

An alternative is to use the POS tagging tools by (Al-Shaibani, 2021).

4.1.6. Phonetic similarity

As illustrated in Table 5, we use the soundex algorithm (Russell and Odell, 1918) to determine whether the source and candidate terms sound the same. This serves a similar purpose to transliteration lexical similarity; however, practicaly characterized with higher precision and lower recall.

4.2. Augmenting the feature space

To provide the machine learning model (Section 5.2) with comparative capability, we augmented the feature space by calculating relative metrics for each candidate against others associated with the same foreign term *occurrence*. Specifically, for three base similarity measures – semantic, translation lexical, and transliteration lexical – we compute a candidate's **rank** and its **score difference** from the top candidate.

Formally, let $x(j)$ denote the score of a base feature $x \in \{\text{semantic}, \text{translation}, \text{transliteration}\}$ for candidate j. Let $\mathscr{C}_i$ represent the set of all candidates belonging to the same term occurrence i, where $i = \text{occ}(j)$.

Source term	Candidate	Phonetically similar
Regavim	ريغافيم (rygAfym - “Regavim”)	True
Lockheed Martin	لوكهيد مارتن (lwkhyd mArtn - “Lockheed Martin”)	True
Lockheed Martin	مثل لوكهيد مارتن (mvl lwkhyd mArtn - “like Lockheed Martin”)	False

Table 5: Phonetic similarity examples

For each candidate j, the augmented features are defined as:

$$\text{Rank}_x(j) = \text{rank}\big(x(j), \mathscr{C}_{\text{occ}(j)}\big)$$

$$\text{Difference}_x(j) = \max_{k \in \mathscr{C}_{\text{occ}(j)}} x(k) - x(j)$$

where $\text{rank}(\cdot)$ denotes the descending rank of the candidate's score $x(j)$ among all candidates within the same occurrence.

Note that we did not augment the feature space for the heuristic approach (Section 5.1), where the comparative capability was obtained from choosing the candidate of maximum score within an occurrence.

5. Results

5.1. Heuristics

Through iterative empirical refinement on the training data, we formulated a composite heuristic score to evaluate each candidate. The total score s for a candidate is defined as the linear combination of five distinct linguistic sub-scores:

$$s = S_L + S_S + S_E + S_P + S_{POS}$$

1. **Lexical Score (S_L):** We prioritize candidates with high lexical similarity to either the translation or the transliteration, applying a threshold $\tau = 0.7$ to filter out low-confidence matches. A slightly higher weight is given to translation similarity:

   $$S_L = 1.2 \cdot l_1 \cdot \mathbb{1}(l_1 \geq \tau) + l_2 \cdot \mathbb{1}(l_2 \geq \tau)$$

   where l_1 and l_2 are the lexical similarities with the translation and transliteration, respectively,

and $\mathbb{1}(\cdot)$ is the indicator function that outputs 1 if the condition is met and 0 otherwise.

2. **Semantic Score (S_S):**

$$S_S = 1.45 \cdot S$$

where S is the semantic similarity score.

3. **Entity Score (S_E):** We reward candidates that share the same named entity type as the source term, provided an entity is actually recognized (denoted by $e \neq \emptyset$):

$$S_E = \begin{cases} 0.7, & \text{if } e \neq \emptyset \text{ and } e = e_s \\ 0.3, & \text{if } e \neq \emptyset \text{ and } e \neq e_s \\ 0, & \text{otherwise} \end{cases}$$

where e and e_s are the entity labels of the candidate and the source term, respectively.

4. **Phonetic Score (S_P):**

$$S_P = \begin{cases} 1, & \text{if phonetically similar} \\ 0, & \text{otherwise} \end{cases}$$

5. **Part-of-Speech Score (S_{POS}):** Because technical terms are predominantly nominal, we explicitly reward nouns and penalize syntactic categories unlikely to form valid terminology:

$$S_{POS} = \begin{cases} 1, & \text{if pos} \in \{\text{noun}, \text{prop\_noun}\} \\ -1, & \text{if pos} \in \{\text{verb}, \text{prep}, \text{conj}\} \\ 0, & \text{otherwise} \end{cases}$$

Finally, for each occurrence of a foreign term, the candidate maximizing the total score s is selected. Table 7 details the performance of this approach.

Note that a first iteration of the heuristics was used to generate a draft of the labels, facilitating the labeling process.

5.2. Machine learning

As shown in Table 2, We annotated 19,405 candidates to build the dataset. Then we trained a binary classifier using the Auto-WEKA package on the Weka Workbench (Frank et al., 2016), which is an automated machine learning (AutoML) tool. We allowed Weka to run for several hours on 8 threads, which resulted in selecting a Random Forest as the classification model, with the arguments displayed in Table 6.

Table 7 summarizes the results. The training set comprises data from nine books, and the test set comprises data from six books. We split the data at the book level to make the task more challenging, where the model is tested on books it has never seen before, which better reflects a real-world scenario.

Component	Configuration
Classifier	`RandomForest`
Arguments	`[-I, 252, -K, 0, -depth, 0]`
Attribute search	`GreedyStepwise`
Search arguments	`[-C, -B, -R]`
Attribute evaluation	`CfsSubsetEval`
Evaluation arguments	`[-L]`
Metric	`errorRate`

Table 6: AutoML configuration and arguments

Category	Metric	Training	Testing
Heuristics	Precision	88.8%	81.5%
	Recall	88.2%	84.4%
	F1 Score	88.5%	82.9%
ML	Precision	100%	90.5%
	Recall	100%	92.4%
	F1 Score	100%	91.4%
Dataset Size		15865	3540

Table 7: Comparison of Heuristics and ML Results

We also tried ranking candidates by partitioning the testing dataset into groups of candidates related to the same occurrence of a term, and ranking them based on the value of their prediction, and the candidate with the highest rank was chosen as the target term (i.e. labeled *True*), and the remaining were labeled *False*. This achieved a 90.1% precision and 92.9% recall.

5.2.1. Feature importance

Table 8 details feature importance measured using Permutation Importance. In general, comparative features (rank and difference) caused significant accuracy drops, confirming that evaluating candidates relative to one another is essential for performance.

5.2.2. Discussion and error analysis

Upon further inspection, it was observed that most of the classification errors made by the ML model involved cases that required expert judgment to discern their correctness. This highlights an advantage of the model: its errors occur primarily on inherently difficult cases, which are challenging even for experts. Table 9 shows an example.

Another cause of errors are peculiar cases where there is one letter difference between the correct translation and the selected candidate, which is caused by Arabic prepositions linked to

Feature	Mean Accuracy Drop
Semantic difference	5.2288%
Part of speech (POS)	2.3701%
Semantic rank	1.7853%
Transliteration lexical difference	1.4831%
Source entity	1.4266%
Translation lexical similarity	1.1808%
Translation lexical difference	1.1610%
Semantic similarity	1.1017%
Transliteration lexical similarity	0.7175%
Translation lexical rank	0.4379%
Transliteration lexical rank	0.3955%
Phonetic similarity	0.2486%
Candidate entity	0.1921%

Table 8: Feature importance ranked by mean accuracy drop (Baseline: 95.96%).

Candidate	Prediction	Label
كتيبة العاصفة (ktybp AlEASfp - "Storm Battalion")	False	True

Table 9: Selecting a wrong candidate for the foreign term *SA* ("Sturmabteilung")

a word. This is usually handled through preprocessing or providing both candidates to the model for it to decide on one, which might fail in some cases. Table 10 shows an example.

5.2.3. Evaluation against a manual glossary

Over 495 books, our extraction pipeline identified 58,570 unique foreign terms. Intersecting these with our manual in-house glossary of 46,787 concepts (a concept here consists of an English, a French, and an Arabic term) yielded 7,368 matches (15.74% of the glossary). This low overlap highlights a major problem in terminology management: assuming in the worst case that a concept was matched with both the English and French foreign terms of our extracted data, we get that at least 43,834 (74.84%) of the extracted terms are novel and need to be handled.

Focusing on the 7,368 *matched concepts*, we evaluated our ranking model's ability to surface the expert Arabic term. The correct glossary entry appeared as the top automated suggestion 30.6% of the time (top-1 accuracy), within the top two automated suggestions 32.99% of the time (top-2 accuracy), and within the top three automated suggestions 33.3% of the time (top-3 accuracy).

Candidate	Prediction	Label
هيئات التوزيع (hy}At AltwzyE - "distribution entities")	True	True
بهيئات التوزيع (bhy}At AltwzyE - "by distribution entities")	True	False

Table 10: Selecting a wrong candidate for the foreign term *Verteilungsstellen* ("distribution points")

By observing these accuracies, we can assume that the percentage of manual Arabic terms in these *matched concepts* that are also present in the 495 processed books to be close to our top-3 accuracy (33.3%). This reasoning is due to the minimal jumps between the top-k accuracies. From this we deduce two key insights:

- About 66.7% of the *matched concepts* have a different Arabic translation by the authors or translators of the processed books than that of the expert. This greatly highlights the inconsistency problem in terminology.
- Among the *matched concepts* with a present Arabic term in the books, our model placed it first in 91.89% of the cases (top-1 accuracy ÷ top-3 accuracy). This indicates that our model effectively elevates the expert translation when it is available. Note that the model performs really well when applying it to problems since it aggregates and compares results from the candidates of all occurrences of a term.

6. Conclusion

This paper introduced the semi-automatic construction of Masrad using Masrad-Ex, a set of tools for extracting and aligning terminology. Our work achieves useful results, contributing to automated *terminology management* and text processing. The created dataset and corpora further enrich the research community, advancing cross-lingual knowledge exchange.

Limitations and future work

Some terms have subtle translations that are inherently complex and challenging to identify, even for human experts. Therefore, the tool is designed to assist rather than replace expert judgment, effectively handling simpler terms automatically while supporting experts with more difficult cases.

As shown in Section 5.2.3, 84.26% of the entries in the manual glossary were not matched. This highlights the richness of the world of terminology and the necessity of drawing on additional sources to expand our termbase. To this end, we plan to apply Masrad-Ex to newly acquired books and expand our tooling to support other types of sources.

To improve the semantic similarity measure 4.1.1, we can fine-tune a sentence transformer (Reimers and Gurevych, 2019) or train a

small model that augments the space of an existing one.

Guided by our feature importance analysis, there is room for stricter evaluation of our feature space, where we can try dropping unimportant features (or improve their quality), and test the addition of new features (e.g. semantic translation similarity).

Ethics statement

The data was collected and used with the appropriate approvals of the intellectual property owners. All results are reported following best academic standards and practices.

7. Bibliographical References

Ahmet Aker, Monica Paramita, and Rob Gaizauskas. 2013. Extracting bilingual terminologies from comparable corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–411, Sofia, Bulgaria. Association for Computational Linguistics.

Nidhal Baccouri. deep-translator.

Ido Dagan and Ken Church. 1994. Termight: Identifying and translating technical terminology. In *Fourth Conference on Applied Natural Language Processing*, pages 34–40, Stuttgart, Germany. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding.

Eibe Frank, Mark A. Hall, and Ian H. Witten. 2016. *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*, fourth edition. Morgan Kaufmann. URL: http://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf.

GokulNC and Prem27. google-transliteration-api.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested arabic named entity corpus and recognition using bert.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Robert C. Russell and Margaret King Odell. 1918. System of soundexing. https://patents.google.com/patent/US1261167A/en.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Awara Sharif. http://awara85.blogspot.com/p/blog-page.html. Accessed: 2024-05-16.

8. Language Resource References

Maged Saeed Al-Shaibani. 2021. *MagedSaeed/farasapy: Python wrapper for the Farasa toolkit*. PID https://github.com/MagedSaeed/farasapy. Accessed: 2025-07-07.

Global WordNet Association. *Arabic WordNet - Arabic Resources*. PID https://globalwordnet.github.io/resources/wordnets-in-the-world. Accessed: 2025-10-25.

Mustafa Jarrar. 2022. *The Arabic Ontology - An Arabic Wordnet with Ontologically Clean Content*. PID https://ontology.birzeit.edu/.

Hossam Mahdy. *Glossary of Arabic Terms for the Conservation of Cultural Heritage in Arabic Alphabetical Order*. PID https://www.iccrom.org/sites/default/files/2017-12/iccrom$_{g}lossary_{e}n - ar.pdf$. Accessed: 2025-10-25.

Nagoudi, El Moatez Billah and Abdul-Mageed, Muhammad and Bouamor, Houda and Habash, Nizar. 2022. *TURJUMAN: A Public Toolkit for Neural Arabic Machine Translation*. PID https://github.com/UBC-NLP/turjuman.

Obeid, Ossama and Zalmout, Nasser and Khalifa, Salam and Taji, Dima and Oudah, Mai and Alhafni, Bashar and Inoue, Go and Eryani, Fadhl and Erdmann, Alexander and Habash, Nizar. 2020. *CAMeL Tools: An Open Source Python*

Toolkit for Arabic Natural Language Processing. European Language Resources Association. PID https://www.aclweb.org/anthology/2020.lrec-1.868.