ArabIE: Joint Entity, Relation and Event Extraction for Arabic

Niama El Khbir, Nadi Tomeh, Thierry Charnois
LIPN, Université Sorbonne Paris Nord - CNRS UMR 7030
Villetaneuse, France
{elkhbir,tomeh,charnois}@lipn.fr

Abstract

Previous work on Arabic information extraction has mainly focused on named entity recognition and very little work has been done on Arabic relation extraction and event recognition. Moreover, modeling Arabic data for such tasks is not straightforward because of the morphological richness and idiosyncrasies of the Arabic language. We propose in this article the first neural joint information extraction system for the Arabic language.

1 Introduction

Information extraction (IE) is the task of identifying and classifying information of interest in a textual document. IE is an important area of research in NLP since it has many practical applications. In this article, we are interested in *joint modeling* of three IE tasks: named entity recognition (NER), relation extraction (RE), and event recognition (ER). A joint multi-tasking system, in comparison to a pipeline system, has the advantage of avoiding the propagation of errors among tasks.

This area of research is well explored in many languages such as English, Chinese, and Spanish. Nguyen and Nguyen (2018) proposed a model that jointly extracts entity mentions, event triggers and event arguments using shared hidden representations in a deep learning framework. Wadden et al. (2019) provided a framework for extracting entities, relations, and triggers using BERT embeddings and graph propagation to capture context relevant for these tasks. Lin et al. (2020) proposed a joint neural framework that extracts entities, relations and events from an input sentence as a globally optimal graph.

For Arabic, however, most proposed models are restricted to NER (Oudah and Shaalan, 2012; Benajiba et al., 2008b). Limited efforts have been dedicated to RE and ER (Taghizadeh et al., 2018; AL-Smadi and Qawasmeh, 2016), and no previous work has addressed them jointly. We attempt to fill this gap in the present work.

Similar to Lin et al. (2020), the model we propose in §2 extracts a graph from an input sequence in two steps: (a) two CRFs (Lafferty et al., 2001) with BIO-based tags are used to identify spans (subsequences of tokens) corresponding to *entities* and *event triggers* (graph nodes); then (b) greedy decoding is used to obtain the output graph.

Since Arabic is morphologically rich (Habash, 2010), entities are not limited to sequences of words like English for instance. Some entities correspond to affixes and some words carry multiple entities. Therefore, modeling on the subword level is necessary. To address this issue, we compare two approaches which we describe in detail in §3. In the first approach, we resort to word tokenization as a preprocessing step. We aim to split morphologically complex words into tokens, each of which corresponds to (or is a part of) one entity at most. An entity can thus be modeled as a sequence of tokens using the standard BIO tags. In the second approach, we augment the BIO tags to encode multiple entities per word, eliminating the need for prior tokenization.

Our contribution in this article is twofold:

- First, we present ArabIE (§2), the first neural joint IE model for Arabic, establishing state-of-the-art results (§4.2) on the ACE 2005 dataset (Walker and Consortium, 2005) (§4.1). We show that the performance of our model is comparable to that of other languages (§4.2).
- Second, we provide an empirical study of the interplay between tokenization (§3) and NER performance and its consequences on RE and ER (§4.2).

2 Multitask Joint Extraction Model

Given a text document as input, we aim at extracting, from each sentence, entities and binary

Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP), pages 331 - 345

relations between them, event triggers, and their arguments. Formally, for an input sequence $\mathbf{x}$ of length L, the information extraction task is the operation that yields, as an output, a graph $G = (V, E)$ whose nodes V are spans of tokens of the input sequence representing identified entities and triggers, and whose edges E represent relations between two entities or event roles (relations between event triggers and their arguments entities). Each node and edge in the graph has a type. Similar to (Lin et al., 2020), our model performs end-to-end IE in four stages.

Token encoding Several combinations of representations from BERT's layers are inspected for encoding the input sequence, as done by Lin et al. (2020) for English data. Ultimately, the input sequence is encoded using the concatenation of BERT's last and third last layers to obtain an embedding for each token, as using these layers improves the performance on most subtasks. Jawahar et al. (2019) showed that BERT last layers contain semantic information about the text, which is beneficial for the processing of Arabic texts. Input sequences are optionally tokenized in a preprocessing step (§3).

Identification Token embeddings are passed to a network composed of a Feed-Forward Network (FFN) layer followed by a Conditional Random Field (CRF) layer. The network labels the sequence using the BIO scheme to identify spans of tokens that correspond to entities or event triggers. We use separate CRF taggers for entities and triggers so that each one specializes in one task. The sequence of labels produced by the CRF encodes a segmentation of the input sequence so that identified entities cannot overlap, the same applies for triggers. On the other hand, entities can overlap with triggers in some cases. The verb أوقفت *(Awqft; she arrested)* is for example a trigger of type `Justice` and the pronoun ت (*t*) is an entity of type `PER`.

Classification At this stage, entities and triggers are identified, but their types are not yet assigned. A fixed-size representation for each span is computed as the average of its first and last token's BERT embeddings. The output is passed to an FFN to obtain a score for each possible type. Again, we use separate FFNs for entities and triggers.

Scoring relations and event roles is performed in a similar manner. An edge between two spans is represented by concatenating their vectors. A relation edge links two entities while a role edge links a trigger to an entity. Representations of edges are passed to an FFN to compute a score for each relation or role type. A special *none* label to indicate the absence thereof. We also use a separate FFN is used for relations and roles.

Decoding We use unconstrained greedy decoding to obtain the output graph: for each node and edge of the graph, we select the highest-scoring type. In our experiments, we tried adding to the graph score a penalty on invalid graph configurations and decode with beam search similar to Lin et al. (2020) but didn't get any improvements.

Training The parameters of all networks are jointly trained end-to-end to minimize the sum of individual task losses. We use the negative log-likelihood of gold BIO paths as a loss function for the CRFs and of the gold label for the FFN classifiers.

3 Subword Entities

As discussed earlier in (§1), a word in Arabic can hold two or more entities anchored on its root or affixes. For example, the word مراسلتنا *(mrAsltnA; our reporter)* comprises two entities: مراسلة *(mrAslp; reporter)* of type person (`PER`) and نا *(nA; our)* of type organisation (`ORG`).[1] This example cannot be handled by our model, which assigns one label to each token in the sequence. Such a mismatch has been considered an anomaly in previous work using sequence labeling approaches (Benajiba et al., 2008a), and subword entities were simply discarded. We propose two solutions to this problem. Figure 1 summarizes the different approaches adopted on the example of the word مراسلتنا.

Word tokenization Subword entities typically correspond to *morphemes*. We, therefore, use a morphological analyzer to tokenize words in context. The probability that each resulting token corresponds to multiple entities decreases dramatically. In practice, we use the analyzer provided by CamelTools (Obeid et al., 2020) and refer to this tokenization scheme by `tok_morph`. The word in the

[1]This example is taken from the ACE 2005 corpus. We use the Buckwalter (Buckwalter) transliteration scheme for Romanization. Note that the taa' marbuuTa (ة; p) transforms to taa' (ت; t) when attached to the suffix (نا; nA).

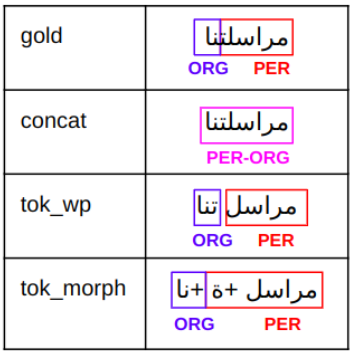

Figure 1: An example of the adopted approaches. Entities are framed in different colors w.r.t their label types. PER: Person, ORG: Organization.

above example is tokenized into three morphemes مراسل +ة +نا *(mrAsl +p +nA)*, the first two tokens correspond to the entity PER, the third one to ORG.

To obtain supervised training data for tokenized sequences, we align each word with its tokens (morphemes) at the character level and use the alignment to project gold entities onto the tokens. An entity is projected onto a token if the majority of its characters align with the token. If multiple entities are projected onto one token, only one of them is randomly selected.

To validate our hypotheses that morphemes are the right level for modeling entities, we compare the morphological analyzer to Word Pieces (Wu et al., 2016), a statistical tokenizer which does not necessarily produce valid affixes. This tokenizer produces مراسل تنا *(mrAsl tnA)* for the example word where the second token is not a morphologically valid suffix and does not exactly match the gold entity نا *(nA; our)*. We refer to this tokenization scheme by tok_wp.

Projection of entities onto tokens is not always perfect either because an entity doesn't correspond to a morpheme in gold data; the tokenizer doesn't produce a valid morpheme; or both. This results in some data loss that we later quantify and take into account during the evaluation phase.

An example of the data loss in tok_wp is that of the sentence سألتها عن الجيران *(s>lthA En AljyrAn; she asked her about the neighbours)*, with ت *(t)* being an entity of type PER, ها *(hA)* an entity of type PER, and الجيران *(AljyrAn)* an entity of type PER, with a relation of type PER-SOC between ها and الجيران. The tok_wp approach yields the following tokens: سأل تها عن الجيران. There is no way to project the two entities ها and ت onto the unique token تها. We therefore randomly project one of the two entities onto this token. If it happens to be ت, then ها is discarded and the PER-SOC relation between ها and الجيران is discarded too. We quantify this data loss in Table 3 for both tokenization schemes.

Label concatenation Instead of tokenization then projection, we concatenate labels of subword entities into one *complex* entity. The example word is thus labeled PER-ORG. This approach is appealing because of its simplicity, but it results in a much larger label set, as some words contain up to four entities. In practice, we restrict the label set to the labels seen in training data. We refer to this tokenization scheme by concat.

4 Experiments and Results

4.1 Experimental setup

Dataset and preprocessing We use the Arabic corpus provided by ACE05[2], which contains different document types annotated with entities, relations and events.

Source	Files	Words	Entities
NW	221	53026	17105
BN	127	26907	9099
WL	55	20181	6234
Total	403	100114	32438
Source	**Relations**	**Triggers**	**Roles**
NW	2674	1270	2957
BN	1606	870	1762
WL	439	130	256
Total	4719	2270	4975

Table 1: General statistics of raw ACE05 data. NW: newswires, BN: broadcast news, WL: weblogs.

The ACE05 data was published in 2006, but very little work has been carried out on it for entity extraction, and no work has been done on relation or event extraction. These previous works are discussed in details in §3.

We randomly split the data into 80% train, 10% dev, and 10% test, as no official split is provided. We will make our splits and our code publicly available.

[2] https://catalog.ldc.upenn.edu/LDC2006T06

Entities	Relations	Triggers	Roles	
FAC: 1427 GPE: 7165 LOC: 1215 ORG: 4885 PER: 17150 VEH: 418 WEA: 481	ART: 338 GEN-AFF: 1142 ORG-AFF: 1379 PART-WHLE: 903 PER-SOC: 643 PHYS: 314	Business: 24 Conflict: 550 Contact: 274 Justice: 379 Life: 398 Movement: 435 Personnel: 152 Transaction: 58	Adjudicator: 91 Agent: 282 Artifact: 378 Attacker: 303 Beneficiary: 22 Buyer: 6 Defendant: 135 Destination: 275 Entity: 584 Giver: 36 Instrument: 266	Origin: 112 Organization: 17 Person: 302 Place: 351 Plaintiff: 12 Prosecutor: 22 Recipient: 17 Seller: 1 Target: 310 Vehicle: 50 Victim: 364

Table 2: Entity, relation, trigger and event role gold ACE05 statistics by label types.

Segmentation We segment each document into sentences using punctuation marks, except for the broadcast news (BN) subcorpus, which we segment into fixed-length sentences due to lack of punctuation. Document segmentation may result in the loss of some entities and triggers (and their associated relations and roles) if a sentence boundary happens to be inside it. Comparing train rows of `gold` and `segm` in Table 3 allows to quantify the data loss after the segmentation phase.

Tokenization Tokenization described in §3 may result in data loss which we quantify in Table 3. However, we use the gold data for dev and test sets for all experiments without discarding any instance.

Dataset Statistics In Table 1, we present statistics done on raw ACE05 files. Note that the difference between role numbers here and gold role numbers of Table 3 is explainable by the fact that we don't handle time roles; arguments that refer to time. We made this choice following Wadden et al. (2019) and Zhang et al. (2019). Thus we also consider that "time" and "value" event arguments are not technically named entities.

In Table 2, we present statistics of entities, relations, triggers and event arguments by label types. Additional statisctis by label subtypes are presented in Tables 9, 10 and 11 of the appendices. In Table 4, we present occurrences of the top 10 most frequent entities of ACE05. The total number of gold entities being 32420, we can easily see that the pronominal entities which are in most cases subwords, are numerous. Hence the need for tokenization to manage them. Note that 21.88% of entities are one-character tokens and 10.18% are two-character tokens.

Tokenization	Split	Entities	Relations	Triggers	Roles
gold	train	26178	3801	1831	3346
	dev	3296	508	235	418
	test	2946	400	204	352
segm	train	26065	3727	1831	3181
concat	train	26065	3727	1831	3181
tok_wp	train	25554	3416	1831	3176
tok_morph	train	25833	3675	1829	3168

Table 3: Statistics on ACE05 train, dev, and test splits. The train, dev, and test sets are identical for all approaches. Comparison of rows `gold` and `segm` show data loss due to document segmentation into sentences, a common pre-tokenization step for all approaches. Comparison of rows `concat`, `tok_wp` and `tok_morph` with row `segm` quantifies data loss due to each tokenization approach.

Training Hyperparameters We trained our model for 80 epochs with a batch size of 6, using BertAdam optimizer, a learning rate of 5e-5 and weight decay of 1e-5 for BERT, and a learning rate of 1e-3 and weight decay of 1e-3 for other parameters.

We used `bert-large-arabertv2` model (Antoun et al., 2020) to conduct all experiments except for `tok_wp` experiments, where we used the `bert-large-arabertv02` tokenizer. Note that the tokenization schemes `tok_morph` do not match the vocabulary of the used BERT model and that there is not yet a BERT adapted to this tokenization procedure. In future works, we aim to solve this mismatch problem by training a BERT language model on the output of the morphological analyzer.

We ran our experiments on an Ubuntu machine, with a GPU Nvidia GEForce RTX 2080 with 8 GB of RAM. We estimated the needed computational budget to 6 GPU hours for each run of each experiment in Table 5.

Entity	Occurrences
ت (t)	2420
ه (h)	1823
ي (y)	1690
ها (hA)	933
هم (hm)	560
وا (wA)	459
نا (nA)	374
الرئيس (Alr}ys; the president)	307
ن (n)	282
أ (>)	279

Table 4: Occurrences of the top 10 most frequent entities of ACE05 gold data.

Evaluation We use precision, recall, and F1 measure for evaluating each task independently. We also combine the individual scores F_1^t of all tasks t into a global (macro) score F_g, where each task is weighted by N_t its number of instances:

$$F_g = \frac{1}{\sum_{t \in \mathcal{T}} N_t} \sum_{t \in \mathcal{T}} N_t F_1^t$$

We consider an entity (resp. trigger) correct if its span and label match those of a gold entity (resp. trigger). Subword entities (§3), however, are allowed not to match exactly their gold span inside the word, they are penalized only if their order inside the word is incorrect. If we take as an example the word of Figure 1, using the `tok_wp` approach, if the model predicts مراسل *(mrAsl)* as an entity of type `PER`, and تنا *(tnA)* as an entity of type `ORG`, the prediction is considered correct. The same evaluation is applied for the `tok_morph` and `concat` approaches.

We consider a relation correct if the participating entities match the gold ones and the relation label matches the gold label. We consider an event role correct if its span and label match the gold one.

While strict evaluation is also possible, we use this approximate approach to emphasize a fair comparison between the tokenization and the concatenation approaches. Both approaches are penalized for the data loss they engender.

4.2 Results

Tables 5 and 6 show results using labels of types (7 entities, 6 relations, 8 triggers, and 22 roles) and subtypes (44 entities, 18 relations, 32 triggers, and 22 roles), for each tokenization scheme. We average the scores across three runs and report numbers for the model with the best average F-score over the four tasks on the dev set.

	`concat`	`tok_wp`	`tok_morph`
Ent.	P: 83.66 ± 0.05 R: 82.26 ± 0.11 F: 82.96 ± 0.03	P: 84.42 ± 0.32 R: 84.05 ± 0.12 F: 84.23 ± 0.22	**P: 85.04 ± 0.25** **R: 85.07 ± 0.2** **F: 85.05 ± 0.12**
Rel.	P: 59.88 ± 1.29 R: 56.88 ± 0.62 F: 58.34 ± 0.94	P: 57.92 ± 1.38 R: 53.0 ± 3.02 F: 55.29 ± 1.67	**P: 62.3 ± 0.42** **R: 63.5 ± 0.61** **F: 62.9 ± 0.51**
Trigg.	P: 67.56 ± 2.38 R: 58.58 ± 0.73 F: 62.74 ± 1.45	**P: 69.49 ± 0.36** R: 57.68 ± 1.89 F: 63.02 ± 1.1	P: 66.32 ± 0.51 **R: 61.11 ± 1.62** **F: 63.59 ± 0.81**
Role	P: 55.8 ± 1.09 R: 43.75 ± 0.85 F: 49.04 ± 0.95	P: 52.75 ± 0.46 R: 40.15 ± 0.81 F: 45.59 ± 0.35	**P: 57.38 ± 1.5** **R: 47.25 ± 0.94** **F: 51.82 ± 0.98**
F_g	76.31	76.66	**78.65**

Table 5: Results on ACE05 data using type labels.

	`concat`	`tok_wp`	`tok_morph`
Ent.	P: 81.86 ± 0.18 R: 80.54 ± 0.32 F: 81.19 ± 0.25	P: 81.74 ± 0.22 R: 80.85 ± 0.13 F: 81.3 ± 0.18	**P: 83.05 ± 0.44** **R: 83.0 ± 0.45** **F: 83.02 ± 0.44**
Rel.	P: 58.61 ± 1.56 R: 55.33 ± 1.33 F: 56.92 ± 1.41	P: 56.62 ± 0.48 R: 51.25 ± 1.0 F: 53.8 ± 0.77	**P: 60.7 ± 0.44** **R: 57.5 ± 0.5** **F: 59.05 ± 0.06**
Trigg.	P: 64.93 ± 2.34 R: 55.88 ± 1.44 F: 60.06 ± 1.76	**P: 66.97 ± 0.68** **R: 56.61 ± 0.25** **F: 61.36 ± 0.14**	P: 64.32 ± 1.38 R: 54.41 ± 1.96 F: 58.96 ± 1.73
Role	P: 53.06 ± 1.07 R: 42.05 ± 1.39 F: 46.9 ± 1.03	P: 50.46 ± 2.45 R: 38.35 ± 0.57 F: 43.56 ± 1.28	**P: 55.48 ± 2.2** **R: 42.61 ± 1.14** **F: 48.2 ± 1.55**
F_g	74.50	74.03	**76.16**

Table 6: Results on ACE05 data using subtype labels.

Existing work on Arabic NER for ACE05 did not address nominal and pronominal entities (Benajiba et al., 2008a) to avoid the tokenization problem, while we handle all grammatical categories of entity mentions.

`tok_morph` results The `tok_morph` approach gets the best F-score on each of the four tasks and has the best F_g score. We suppose that morphological information introduced by the tokenizer helps the model to improve the recognition of relations and roles.

`concat` results The `concat` approach gets the lowest F_g score. We can notice that its performance on triggers using type labels is quite close to that of `tok_morph`, but its performance on entities is poor compared to `tok_wp` and `tok_morph` approaches. We explain this by the increase in the number of labels to classify in this approach; 24 entity type labels (resp. 127 entity subtype labels), such as `PER-VEH`, `ORG-VEH`, `VEH-VEH` (resp. `PER:Group-VEH:Air`,

PER:Individual-VEH:Air), instead of 7 entity type labels (resp. 44 entity subtype labels), such as PER, LOC, VEH... (resp. PER:Group, PER:Individual, VEH:Air...) for the other approaches.

Relations (resp. roles) F-score is degraded by 4.56 (resp. 2.78) points compared to that of tok_morph even if the relation labels number is the same for these two approaches. We explain this by the fact that when the classification and identification of entities become more complex, the part of the loss specific to entities becomes difficult to minimize, which forces the model to prioritize this task over the others, thus degrading relation and role performance.

tok_wp results Entity and relation performance of tok_wp is close to that of tok_morph and better than that of concat. However, this approach gets the lowest F-score for relation and role tasks. This is partly due to a larger number of discarded entities in this approach than in the other approaches. More discarded entities leads to more discarded relations, and since we penalize each model with respect to discarded instances, this explains the discrepancy in performance.

Type labels experiments details We present in this subsection score details of the experiments of Table 5. Table 7 shows entity, relation, trigger, and role scores by type labels.

We do not report scores details of the subtype label experiments (Table 6) because they are too numerous, and in general the behavior and the performance of the subtype labels experiments follow that of the type label experiments.

We notice that among the entity types, PER has the best F-score. Likewise, among the relation types, ORG-AFF has the best F-score. PER and ORG-AFF represent respectively 52.87% and 29.22% of the total number of entities and relations.

Imbalanced Data Problem We notice furthermore that Business events have an F-score of 0; they represent only 0.5% (of the total number of events), which is a limited amount of data to train the model to recognize this class. The same behavior (with an F-score of 0) is observed for role types Beneficiary, Buyer, Organization, Prosecutor, Recipient, and Seller as they represent respectively 0.14%, 0.41%, 0.53%, 0.41%, and 0.02% of the total number of roles. For example, the Recipient role is always incorrectly predicted by the model as the Beneficiary role, since these two roles are very close semantically in the context of a Transaction event.

Comparison to other languages Table 8 show state-of-the-art F-scores of joint IE with ACE05 dataset for different languages. English, Chinese, and Spanish experiments were borrowed from Lin et al. (2020), who trained their model with type labels for entity, relation, and roles, and with subtype labels for triggers. We thus give scores of Arabic following this pattern.

Overall results Unless using concat tokenization procedure, our model assigns one label to each input token, which establishes an upper bound on its performance since multi-label tokens are out of its reach. For example, p+drop experiments could at most reach a recall of 97.31 for entities, 90.75 for relations, and 93.46 for roles; i.e., at most an F-score of 98.63 for entities, 95.15 for relations, and 96.71 for roles.

Importantly, the performance of our three systems of Table 5 is comparable to other languages (Lin et al., 2020) (details in Table 8).

Since there was no baseline addressing the entirety of ACE05 entities, nor a system for RE and ER, we propose tok_morph as a baseline.

5 Error Analysis

Error analysis is important to understand the model's weaknesses and to attempt to fix them in future work. Thus, we examined a sample of 32 sentences where we found 110 remaining errors from experiments with tok_morph tokenization and type labels.

Entity Errors About 23% are errors related to pronominal entities; these errors either come from entities predicted by the model and not annotated in the gold data or vice-versa or from correctly identified entities but incorrectly classified. For example, in the word صادرتها (*SAdrthA; confiscated it*), the pronoun ت (*t*) is annotated in gold data as a PER entity that the model does not predict. These errors are most likely due to the lack of labeling of a considerable number of pronominal entities of the gold data. As example, for the word المسلحين (*Alm-slHyn; armed*), the model predicts the pronoun ين (*yn*) as a PER entity but it's not annotated in the gold data, although this pronoun was annotated

Entities	Relations	Triggers	Roles	
FAC: 0.82 ± 0.0 GPE: 0.85 ± 0.0 LOC: 0.66 ± 0.02 ORG: 0.76 ± 0.0 PER: 0.9 ± 0.0 VEH: 0.78 ± 0.01 WEA: 0.81 ± 0.03	ART: 0.58 ± 0.02 GEN-AFF: 0.62 ± 0.02 ORG-AFF: 0.73 ± 0.01 PART-WHLE: 0.56 ± 0.01 PER-SOC: 0.63 ± 0.02 PHYS: 0.31 ± 0.07	Business: 0.0 ± 0.0 Conflict: 0.67 ± 0.01 Contact: 0.39 ± 0.02 Justice: 0.62 ± 0.02 Life: 0.84 ± 0.0 Movement: 0.42 ± 0.06 Personnel: 0.57 ± 0.03 Transaction: 0.71 ± 0.02	Adjudicator: 0.37 ± 0.03 Agent: 0.44 ± 0.04 Artifact: 0.6 ± 0.04 Attacker: 0.55 ± 0.02 Beneficiary: 0.0 ± 0.0 Buyer: 0.0 ± 0.0 Defendant: 0.22 ± 0.06 Destination: 0.58 ± 0.05 Entity: 0.41 ± 0.0 Giver: 0.35 ± 0.11 Instrument: 0.69 ± 0.04	Origin: 0.42 ± 0.0 Organization: 0.0 ± 0.0 Person: 0.57 ± 0.04 Place: 0.49 ± 0.03 Plaintiff: 0.11 ± 0.16 Prosecutor: 0.0 ± 0.0 Recipient: 0.0 ± 0.0 Seller: 0.0 ± 0.0 Target: 0.5 ± 0.05 Vehicle: 1.0 ± 0.0 Victim: 0.67 ± 0.04

Table 7: Entity, Relation, Trigger and Role F-score details of experiment of Table 5 using `tok_morph` approach and type labels.

Language	Ent.	Rel.	Trigg.	Role
English	89.6	58.6	72.8	54.8
Chinese	88.5	62.4	65.6	52.0
Spanish	81.3	48.1	56.8	40.3
Arabic	85.05	62.9	58.96	51.82

Table 8: State-of-the-art F-scores of joint IE for different languages. Arabic scores are those of `tok_morph` experiments.

167 times in words like المتقاعدين (*AlmtqAEdyn; retirees*), الآخرين (*AlAxryn; the others*), and الراغبين (*AlrAgbyn; willing to*). Note that pronominal entities represent 31% of the total gold entities.

Relation Errors About 14% of the remaining errors are multiple relation entities, i.e., relations incorrectly predicted because their entities are involved in multiple relations. For example, in the gold annotations of the sentence وزير العدل المصري (*wzyr AlEdl AlmSry; Egyptian Minister of Justice*), the word وزير (*wzyr; Minister*) is involved in two relations of types `ORG-AFF` (resp. `GEN-AFF`) with the word العدل (*AlEdl; Justice*) (resp. المصري (*AlmSry; Egyptian*)). The model only predicts the first `ORG-AFF` relation between the two first words.

At least 6% are correctly identified and incorrectly classified relations, i.e., the model correctly predicts the two participating entities of the relation but incorrectly predicts the relation type. This error is usually due to the ambiguity induced by the existing semantic proximity between some relation types, such as `PART-WHOLE` and `ORG-AFF`.

Events Errors Nearly 23.5% are annotation errors, particularly related to triggers and roles. Specifically, out of the 35 remaining event errors, 67% are related to annotation omissions. As an example, in the sentence اتصل به شقيقيه (*AtSl bh $qyqyh; his brothers called him*), the model predicts the verb اتصل (*AtSl; called*) as a trigger of type `Contact`. This trigger is not annotated in the gold data but the model's prediction seems correct because an event of type `Contact` is defined in the annotation guide by: explicit phone or written communication between two or more parties. In the annotation guide the verb called in the sentence "John called Jane last night" is given as an example of a trigger of type `Contact`. Figure 2 presents a recurring example of a long sentence containing several omitted roles. In this sentence, we distinguish three errors: (1) the word المتهمين (*Almthmyn; The accused*) is predicted as an `Agent` argument by the model, which is intuitively correct as an `Agent` is defined in the annotation guide by "the attacking agent or the one that enacts the harm". This word is incorrectly annotated in the gold sentence as an argument of type `Victim`. (2) The word رفاق (*rfAq; companions*) is predicted as an argument of type `Agent` which is intuitively correct. This word is not annotated in the gold sentence as an argument. (3) The word الصائغ (*AlSAg; the jeweler*) is predicted as arguments of type `Victim` which is intuitively correct as a `Victim` is defined in the annotation guide by: the person who died. This word is not annotated in the gold sentence.

6 Related work

Entity Extraction Most Arabic IE work focuses on NER. We cite (Naji, 2012), who used artificial neural networks for NER. (Oudah and Shaalan, 2012) tested a hybrid approach, including both rule-

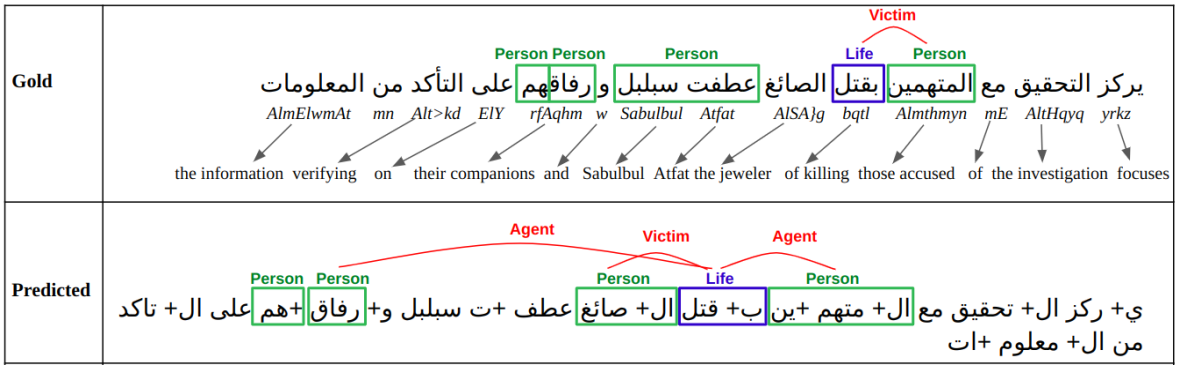

Figure 2: An example of remaining event errors (annotation omissions), using `tok_morph` tokenization and type labels. Entities are framed in green, triggers are framed in blue, event arguments (roles) are represented by red edges. `ORG: Organization, GPE: Geo-Political Entity, PER: Person.`

based and machine learning approaches. (Benajiba et al., 2008b) proposed an SVM-based model with a combination of language-dependent and language-dependent features, showing the relevance of morphological features for rich languages like Arabic. (Benajiba et al., 2010) built a system augmented by deeper lexical, syntactic, and morphological features that were extracted from noisy data obtained via projection from an Arabic-English parallel corpus. (Helwe et al., 2020) proposed a semi-supervised learning approach to train a BERT-based NER model using labeled and semi-labeled datasets. The works that deal with NER using ACE05, ACE04, or ACE03 either preprocess the data differently from ours, which results in a very different number of entities than ours or use different entity types than the one we used. For example, Benajiba et al. (2008b) evaluate their model separately for each data type of ACE05 (NW, BN, WL). In addition, they remove all annotations that they consider not oriented to the entity detection and recognition tasks, such as the nominal and pronominal entities, and only keep the named ones, which leads them to a total number of entities in the training and test corpora of 10218. This makes their performance incomparable to ours because we evaluate the model with almost 32000 entities for all our proposed approaches. Other work (Benajiba et al., 2010, 2009, 2008a) same preprocessing of Benajiba et al. (2008b). Oudah and Shaalan (2012) tested their model performance on `Date, Time, Price, Measurement, and Percent` entities of ACE05, while we test our model on the principal entity types (`PER, LOC, ORG, FAC, VEH`...).

Relation Extraction Arabic RE works include (Mohamed et al., 2015), who proposed a distant supervised learning model with specific features that characterize Arabic relations. (Sarhan et al., 2016) presented a semi-supervised pattern-based bootstrapping technique for RE using stemming and semantic expansion. (Taghizadeh et al., 2018) used a combination of kernel functions and the universal dependency parsing for supervised relation extraction. We can't compare our work to these as relation extremities (entities) are already recognized in a NER pre-processing, while we extract all information jointly.

Event Extraction Very little work has been done on ER; (AL-Smadi and Qawasmeh, 2016) proposed a knowledge-based approach for ER on Arabic tweets. And (Alsaedi and Burnap, 2015) proposed a classification/ clustering-based framework to detect real-world events from Twitter. (Ahmad et al., 2020) developed a Graph Attention Transformer Encoder to generate structured contextual representations for cross-lingual relation and event extraction working on ACE05. Yet, they haven't addressed the problem of the mismatch between the tokenization and the annotations; problematic entities were simply discarded.

7 Conclusion

We presented the first joint IE model for Arabic and showed a comparable performance to other languages. We also proposed two approaches to address subword entities, a situation specific to morphologically rich languages including Arabic, and showed that morphological information is important to their recognition. Our hope is that our work will provide a strong baseline for further research and increase interest in IE tasks which remain understudied by the Arabic NLP community.

Acknowledgements

This work is partially supported by a public grantoverseen by the French National Research Agency (ANR) as part of the program Investissements d'Avenir (ANR-10-LABX-0083).

References

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2020. GATE: graph attention transformer encoder for cross-lingual relation and event extraction. *CoRR*, abs/2010.03009.

Mohammad AL-Smadi and Omar Qawasmeh. 2016. Knowledge-based approach for event extraction from arabic tweets. *International Journal of Advanced Computer Science and Applications*, 7(6).

Nasser Alsaedi and Pete Burnap. 2015. Arabic event detection in social media. In *Computational Linguistics and Intelligent Text Processing*, pages 384–401, Cham. Springer International Publishing.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Yassine Benajiba, Mona Diab, and Paolo Rosso. 2008a. Arabic named entity recognition using optimized feature sets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, page 284–293, USA. Association for Computational Linguistics.

Yassine Benajiba, Mona Diab, and Paolo Rosso. 2009. Using language independent and language specific features to enhance arabic named entity recognition. *Int. Arab J. Inf. Technol.*, 6:463–471.

Yassine Benajiba, Mona T. Diab, and P. Rosso. 2008b. Arabic named entity recognition: An svm-based approach.

Yassine Benajiba, Imed Zitouni, Mona Diab, and Paolo Rosso. 2010. Arabic named entity recognition: Using features extracted from noisy data. pages 281–285.

Tim Buckwalter. *Arabic Morphological Analyzer Version 2.0 LDC2004L02*. Linguistic Data Consortium, 2004.

Nizar Y. Habash. 2010. *Introduction to Arabic natural language processing*, 1 edition, volume 3 of *Synthesis Lectures on Human Language Technologies*. Morgan and Claypool Publishers.

Chadi Helwe, Ghassan Dib, Mohsen Shamas, and Shady Elbassuoni. 2020. A semi-supervised BERT approach for Arabic named entity recognition. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 49–57, Barcelona, Spain (Online). Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of The 58th Annual Meeting of the Association for Computational Linguistics*.

Reham Mohamed, Nagwa M. El-Makky, and Khaled Nagi. 2015. Arabrelat: Arabic relation extraction using distant supervision. In *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, IC3K 2015, page 410–417, Setubal, PRT. SCITEPRESS - Science and Technology Publications, Lda.

Nazlia Naji. 2012. Arabic named entity recognition using artificial neural network. *Journal of Computer Science*, 8:1285–1293.

Trung Minh Nguyen and Thien Huu Nguyen. 2018. One for all: Neural joint modeling of entities and events.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Mai Oudah and Khaled Shaalan. 2012. A pipeline Arabic named entity recognition using a hybrid approach. In *Proceedings of COLING 2012*, pages 2159–2176, Mumbai, India. The COLING 2012 Organizing Committee.

Injy Sarhan, Yasser El-Sonbaty, and Mohamad Abou El-Nasr. 2016. Semi-supervised pattern based algorithm for arabic relation extraction. *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 177–183.

Nasrin Taghizadeh, Heshaam Faili, and Jalal Maleki. 2018. Cross-language learning for arabic relation extraction. *Procedia Computer Science*, 142:190–197.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

C. Walker and Linguistic Data Consortium. 2005. *ACE 2005 Multilingual Training Corpus*. LDC corpora. Linguistic Data Consortium.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint Entity and Event Extraction with Generative Adversarial Imitation Learning. *Data Intelligence*, 1(2):99–120.

A Additional Dataset Statistics

We present here additional statistics of ACE05 subtype labels used in experiments of Table 6. We provide statistics of time roles even if we do not handle them. Note that event arguments (roles) do not have subtype labels.

Entity types	Number	Number by subtype	Percentage
Person	17150	Group: 6572 Individual: 10523 Indeterminate: 55	52.87%
Organization	4885	Media: 821 Commercial: 591 Government: 1432 Non-Governmental: 1171 Sports: 649 Educational: 135 Medical-Science: 22 Religious: 29 Entertainment: 36	15.06%
Location	1215	Boundary: 147 Celestial: 79 Region-General: 597 Region-International: 211 Land-Region-Natural: 89 Water-Body: 74 Address: 18	37.45%
Geographical/Social/Political	7165	Population-Center: 1328 Nation: 4560 Continent: 112 Special: 718 GPE-Cluster: 141 County-or-District: 146 State-or-Province: 160	22.09%
Facility	1127	Path: 176 Building-Grounds: 727 Airport: 23 Subarea-Facility: 117 Plant: 84	3.47%
Vehicle	418	Land: 185 Subarea-Vehicle: 2 Water: 76 Air: 155	12.87%
Weapon	481	Projectile: 179 Underspecified: 105 Sharp: 5 Shooting: 111 Blunt: 16 Exploding: 45 Chemical: 10 Nuclear: 10	14.83%
Total	32438	32438	100%

Table 9: Statistics of ACE05 entity types and subtypes.

Relation types	Number	Number by subtype	Percentage
Gen-Affiliation	1142	Org-Location: 561 Citizen-Resident-Religion-Ethnicity: 581	24.20%
Org-Affiliation	1379	Employment: 1136 Sports-Affiliation: 24 Membership: 195 Student-Alum: 13 Ownership: 6 Founder: 3 Investor-Shareholder: 2	29.22%
Part-Whole	903	Geographical: 607 Subsidiary: 291 Artifact: 5	19.13%
Personal-Social	643	Business: 306 Lasting-Personal: 81 Family: 256	13.62%
Physical	314	Located: 263 Near: 51	6.65%
Agent-Artifact	338	User-Owner-Inventor-Manufacturer: 338	7.16%
Total	4719	4719	100%

Table 10: Statistics of ACE05 relation types and subtypes.

Event types number	Event subtypes number	Roles number	Total Roles
Life: 398	Be-Born: 6	Person: 6 Place: 1 Time-Before: 1 Time-Within: 2	10
	Marry: 16	Place: 2 Person: 20 Time-Within: 2 Time-Holds: 1 Time-After: 1	26
	Divorce: 5	Person: 7 Place: 1	8
	Injure: 127	Victim: 125 Place: 52 Instrument: 46 Agent: 32 Time-At-Beginning: 1 Time-Within: 23 Time-After: 1	280
	Die: 244	Victim: 239 Agent: 83 Place: 78 Instrument: 53 Time-Within: 55 Time-Starting: 6 Time-Ending: 1 Time-At-Beginning: 1 Time-At-End: 3 Time-Holds: 1 Time-Before: 2	522
Movement: 435	Transport: 435	Artifact: 369 Origin: 111 Destination: 271 Agent: 96 Vehicle: 51 Time-Before: 5 Time-After: 2 Time-Within: 83 Time-Starting: 12 Time-Ending: 3 Time-At-Beginning: 3 Time-At-End: 1 Time-Holds: 6	1013
Transaction: 58	Transfer-Ownership: 10	Buyer: 6 Seller: 1 Beneficiary: 3 Artifact: 9 Price: 1 Place: 1 Time-Holds: 2 Time-Within: 3	26
	Transfer-Money: 48	Money: 33 Giver: 36 Recipient: 17 Beneficiary: 19 Place: 7 Time-Starting: 1 Time-Within: 11 Time-Holds: 1	125

Event types number	Event subtypes number	Roles number	Total Roles
Business: 24	Start-Org: 14	Org: 11 Agent: 14 Place: 3 Time-Before: 1 Time-Within: 1	30
	Merge-Org: 1	Org: 1	1
	Declare-Bankruptcy: 1	Org: 1	1
	End-Org: 8	Org: 4 Agent: 1 Place: 2 Time-Starting: 1 Time-Within: 1 Time-Holds: 1	10
Conflict: 550	Attack: 477	Attacker: 304 Target: 313 Instrument: 168 Place: 174 Time-Starting: 10 Time-At-Beginning: 2 Time-Within: 88 Time-After: 3 Time-Holds: 5 Time-Before: 2	1069
	Demonstrate: 73	Entity: 57 Place: 35 Time-Before: 1 Time-Starting: 1 Time-Within: 17 Time-Holds: 3	114
Contact: 274	Meet: 217	Entity: 362 Place: 91 Time-Starting: 7 Time-At-Beginning: 1 Time-Before: 1 Time-Within: 69 Time-Holds: 6 Time-Ending: 1 Time-After: 1	539
	Phone-Write: 57	Entity: 97 Place: 5 Time-Within: 8 Time-After: 1	111
Personnel: 152	Start-Position: 46	Entity: 12 Person: 44 Position: 7 Place: 12 Time-Before: 1 Time-Starting: 2 Time-Holds: 1 Time-Within: 9	88
	End-Position: 58	Entity: 6 Person: 55 Position: 3 Place: 9 Time-Within: 9 Time-Holds: 3 Time-Ending: 2	87
	Nominate: 7	Person: 7 Agent: 4 Position: 1 Place: 1 Time-Within: 1	14
	Elect: 41	Entity: 15 Person: 27 Position: 4 Place: 9 Time-Starting: 2 Time-Within: 10	67

Event types number	Event subtypes number	Roles number	Total Roles
Justice: 379	Arrest-Jail: 109	Person: 99 Agent: 49 Place: 32 Crime: 15 Time-Before: 3 Time-Starting: 8 Time-Within: 26 Time-Holds: 6 Time-Ending: 1 Time-After: 1 Time-At-End: 1	241
	Release-Parole: 31	Entity: 13 Person: 31 Place: 5 Time-Before: 1 Time-Within: 13	63
	Trial-Hearing: 65	Crime: 8 Defendant: 41 Adjudicator: 21 Prosecutor: 10 Place: 4 Time-Before: 1 Time-Starting: 5 Time-Within: 13 Time-Holds: 1 Time-After: 1	105
	Charge-Indict: 52	Defendant: 47 Prosecutor: 12 Adjudicator: 15 Crime: 21 Place: 4 Time-Before: 1 Time-Starting: 1 Time-At-Beginning: 1 Time-Within: 9 Time-Holds: 1 Time-Ending: 1	113
	Sue: 2	Adjudicator: 2 Time-Within: 1	3
	Convict: 5	Defendant: 5 Crime: 4 Place: 1 Adjudicator: 3 Time-Within: 2	15
	Sentence: 51	Adjudicator: 22 Defendant: 36 Sentence: 37 Crime: 20 Place: 4 Time-Starting: 5 Time-Within: 9 Time-Holds: 5 Time-Ending: 5	143
	Fine: 33	Entity: 28 Adjudicator: 12 Money: 41 Crime: 5 Place:1 Time-Within: 4	91
	Extradite: 7	Person: 7 Origin: 1 Destination: 4 Agent: 3	15
	Acquit: 3	Defendant: 3 Adjudicator: 1	4
	Appeal: 19	Adjudicator: 16 Plaintiff: 12 Crime: 2 Defendant: 1 Place: 1 Time-Within: 5 Time-Ending: 1	38
	Pardon: 2	Defendant: 2 Place:1	3

Table 11: Statistics of ACE05 trigger types and subtypes and role types.