Improving Named Entity Recognition in Chinese Abstract Meaning Representation

Nicholas A Miller

2018 年 4 月 24 日

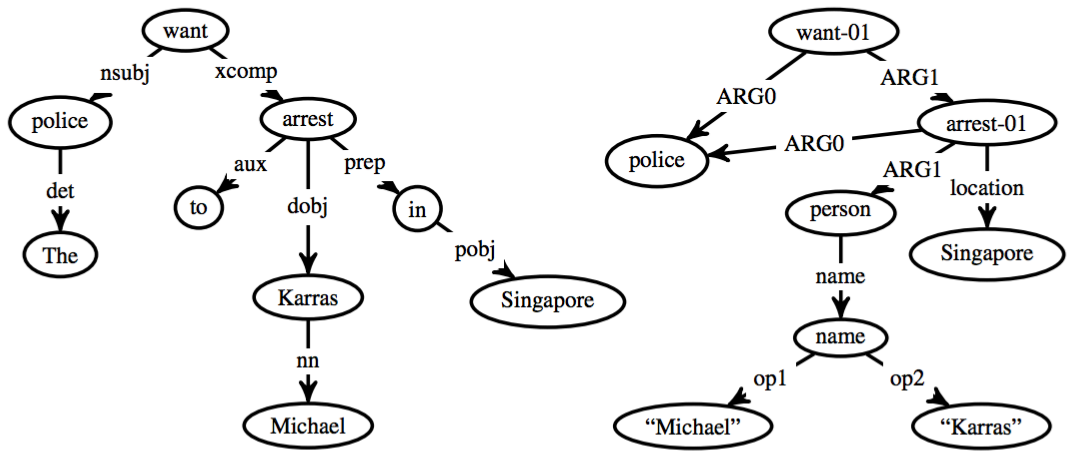
Abstract

In this project, I examine several strategies for improving handling of named entities in Chinese Abstract Meaning Representation, using the CAMR parser as described in Wang et al. (2016). Strategies include normalization of NE tags, and training the model with various combinations of features. Overall, the approach with the best performance was to normalize NE tags in a preprocessing step before training the model. This approach resulted in the named entity F1 score (Damonte et al., 2016) increasing from .57 to .59.

1 Abstract Meaning Representation

Abstract meaning representation (AMR) is a formalism that represents sentences as directed graphs, with the intent of abstracting away from language-specific surface distinctions such as word order, part of speech, tense, aspect, and morphological variation.

In this project, I use the CAMR parser, a transition-based parser (Wang et al., 2015), which employs a pipeline to convert dependency graphs to AMR graphs. Figure 1, reproduced from Wang et al. (2015) shows a sample conversion from dependency graph to AMR graph. In particular, the transition-based model in this version of CAMR uses the transition actions described in Table 1 to derive AMR graphs from dependency trees (for a detailed explanation of these transition actions, see Wang et al. 2015 and Wang et al. 2016).



(a) Dependency tree (b) AMR graph

Figure 1: Example transition from dependency tree to AMR graph, from Wang et al. (2015b)

2 Named Entities in AMR

The CAMR parser uses the Stanford Named Entity (NE) Tagger as part of its pipeline, and learns to convert the Stanford NE tags to AMR NE tags. For Chinese, the inventory of named entities in AMR is much richer than the inventory of Stanford NE tags. Stanford NE tags for Chinese include ‘LOCATION’, ‘PERSON’, ‘FACILITY’, ‘DEMONYM’, ‘MISC’, ‘GPE’, and ‘ORGANIZATION’. In contrast, AMR guidelines allow for the full range of named entities described in Figure 2, plus ‘thing’ in the case of generic named entities that do not fit one of the above. In addition to the named entity types in Figure 2, the guidelines also allow annotators to create additional named entity types when warranted by the context.[1]

3 Data and Tools

The data consist of a pre-release version of a Chinese AMR corpus consisting of 10,150 sentences extracted from the Chinese Treebank (CTB) (Xue et al., 2005). [2] First, the data was split into training, development, and test set of 9,093, 413, and 494 instances respectively. The parser uses the Stanford CoreNLP toolkit (Manning et al., 2014) to perform segmentation, part-of-speech tagging, and named entity tagging, and the parser described in Wang & Xue (2014) to perform phrase structure

[1] see https://www.isi.edu/ ulf/amr/lib/ne-types.html

[2] see http://www.cs.brandeis.edu/ clp/camr/camr.html for more information

Action	Definition
MERGE	merge several nodes into a single node. Used to consolidate entity names to a single node.
DELETE-NODE	completely remove a node. Used to remove function words with no outgoing arcs.
REPLACE-HEAD	remove node from s-buffer and replace with node from b-buffer. Used to remove nodes with no associated AMR concept (e.g. function words) which have children.
SWAP	switch the position of two nodes. Used when a node is the head in a dependency tree but not in an AMR graph or vice-versa (e.g. conjunctions such as 'and').
REATTACH	attach a node to a new head. Used to rearrange nodes after a SWAP action.
REENTRANCE	link node to other nodes which may be its parent.
INFER	create a new concept node and make corresponding attachments. Used for AMR concepts with no corresponding natural language lemmata.

Table 1: CAMR parser transition actions

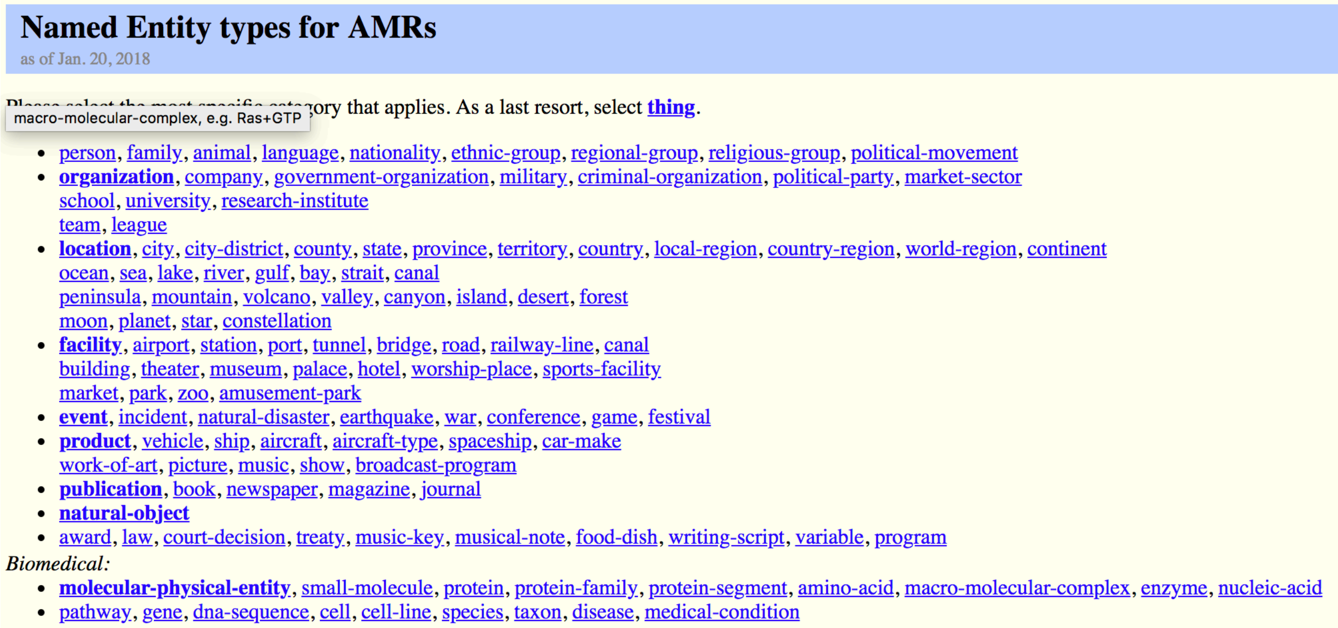
and dependency parsing.

4 Evaluating Chinese NE handling in CAMR

Our first evaluation measure is the fine-grained variation of the Smatch score (Cai & Knight, 2013) described in Damonte et al. (2016) which provides a precision/recall/F1 measure for named entities. [3] In addition to this Smatch variation, I introduce sentence-by-sentence fine-grained evaluation of NE tagging. The evaluator iterates over each sentence, compares the lists of named entities between the gold and parsed graphs, and counts the following:

- **Extra:** the parser adds a NE tag where the gold has none
- **Missing:** the parser lacks a NE tag where the gold has one
- **Mismatch:** the number of NE tags matches, but the tags themselves are different
- **Match:** the number of NE tags matches (and is nonzero), and the tags are all the same

[3] https://github.com/mdtux89/amr-evaluation



Named Entity types for AMRs

as of Jan. 20, 2018

Please select the most specific category that applies. As a last resort, select **thing**.

macro-molecular-complex, e.g. Ras+GTP

- person, family, animal, language, nationality, ethnic-group, regional-group, religious-group, political-movement
- **organization**, company, government-organization, military, criminal-organization, political-party, market-sector
  school, university, research-institute
  team, league
- **location**, city, city-district, county, state, province, territory, country, local-region, country-region, world-region, continent
  ocean, sea, lake, river, gulf, bay, strait, canal
  peninsula, mountain, volcano, valley, canyon, island, desert, forest
  moon, planet, star, constellation
- **facility**, airport, station, port, tunnel, bridge, road, railway-line, canal
  building, theater, museum, palace, hotel, worship-place, sports-facility
  market, park, zoo, amusement-park
- **event**, incident, natural-disaster, earthquake, war, conference, game, festival
- **product**, vehicle, ship, aircraft, aircraft-type, spaceship, car-make
  work-of-art, picture, music, show, broadcast-program
- **publication**, book, newspaper, magazine, journal
- **natural-object**
- award, law, court-decision, treaty, music-key, musical-note, food-dish, writing-script, variable, program

Biomedical:

- **molecular-physical-entity**, small-molecule, protein, protein-family, protein-segment, amino-acid, macro-molecular-complex, enzyme, nucleic-acid
- pathway, gene, dna-sequence, cell, cell-line, species, taxon, disease, medical-condition

Figure 2: AMR named entity types, from https://www.isi.edu/ ulf/amr/lib/ne-types.html

This fine-grained evaluation is preliminary and imperfect. In particular, it assumes that if a concept receives a NE tag in the gold version, it will also get a NE tag in the parsed version. In other words, it does not account for the possibility that the gold and parsed AMR graphs have matching NE tags, but the tags themselves were assigned to different concepts. In practice, however, this issue did not appear to come up frequently in the test data. Thus, this is a useful first step towards examining the fluctuations in NE handling performance.

5 Approaches to improving Chinese NE handling in CAMR

A description of the approaches I tried appears in Table 2. The 'basic' approach uses just the basline CAMR features shown in Table 3 (for a detailed description of the features, see Table 4 and also e.g. Wang et al. 2015).

'+postproc' refers to the addition of post-processing to normalize some of the NE tags. Manual examination of the data revealed some inconsistencies in spelling, as well as some inconsistencies in whether NE tags were given in English or Chinese. With that in mind, I normalized such discrepencies during postprocessing. For example, Chinese NE tags such as 国家 *guojia* 'country' and 国 *guo* 'country' are normalized to the English NE tag `country`, as are misspelled English NE tags such as `coutry` and `coountry`.

Approach	Description
Basic	AMR parsing with baseline features
Basic+postproc	+postprocessing to normalize NE tags (e.g. country/coutry/coountry/国家/国 → country)
Sibling unigram	+unigram features added to target siblings in the dependency tree
Sibling unigram+postproc	+postprocessing to normalize NE tags
Sibling bigram	+bigram features added to target siblings in the dependency tree
Sibling bigram+postproc	+postprocessing to normalize NE tags
Preproc+retrained	Normalize NE tags then retrain model with baseline + sibling unigram + sibling bigram features

Table 2: Approaches to improving Chinese AMR NE handling

'Sibling unigram' and 'Sibling bigram' refers to retraining the CAMR parser model after adding features which target unigram and bigram features of the dependency trees (Table 5).

Finally, for 'Preproc+retrained', I applied the named entity normalization described in postprocessing above, except in this case, I normalized the entity tags *before* training the model, and then trained the model using the sibling bigram features described above.

Feature Type	CAMR abbreviation
unigram	*s0_ne tx, s0_w tx, s0_lemma tx, s0_t tx, s0_dl tx, s0_len tx, b0_ne, b0_w, b0_lemma, b0_t, b0_dl, b0_len, a0_ne, a0_w, a0_lemma, a0_t, a0_dl*
parent/child	*s0_nech, s0_lemma s0_nech, s0_c1lemma tx, s0_c1dl s0_c1lemma, s0_cpt s0_p1_ne s0_c1lemma, s0_p1_ne, s0_p1_w, s0_p1_lemma, s0_p1_t, s0_p1_dl*
sibling	*b0_lemma b0_rsb_dl*
action	*b0_reph, b0_lemma b0_nswp*
path	*b0_pathpwd b0_lemma s0_lemma, b0_apathpwd a0_lemma b0_lemma, b0_pathpwd, b0_apathpwd*
distance	*dist1, dist1 b0_pathp, dist2, dist2 b0_apathp*
bigram	*s0_lemma b0_t, s0_lemma b0_dl, s0_t b0_lemma, s0_dl b0_lemma, s0_ne b0_ne, a0_t b0_lemma, a0_dl b0_lemma, a0_ne b0_ne, a0_cpt b0_cpt, a0_cpt b0_cpt s0_lemma, a0_cpt b0_ne*

Table 3: CAMR baseline features

Abbreviation	**Meaning**
a0	top of a-buffer
b0	top of b-buffer
c1	first child
cpt	concept
dl	dependency relations
len	length
ne	named entity
nswp	number of swaps
p1	first parent
reph	rephrase count
rsb	right sibling
s0	top of s-buffer
tx	transition
w	word

Table 4: Key to CAMR feature abbreviations

6 Results

The results appear in Table 6. We can see that overall, the number of matches tends to increase with the incorporation of additional features, while the number of missing, extra, and mismatched NE tags tends to fluctuate. We can also observe that postprocessing tends to decrease mismatches and increase matches, albeit only modestly. Overall, the best-performing approach was preprocessing and retraining with sibling bigram features. This was the only approach to actually increase the Smatch F1 score for NEs, and led to across-the-board improvements: compared to the baseline, we can see decreased 'extra', 'missing', and 'mismatch', and increased 'match', all of which are improvements over the baseline.

The results in Table 6 also allow us to make some other general observations. The most common type of NE error appears to be 'Extra', in which the parser posits named entity tags where the gold annotation had none. In comparison, 'Mismatch' errors are relatively few. One might suspect that the more fine-grained AMR NE tags could be difficult to learn from the more limited Stanford NE tags. However, if error propagation from the Stanford NE tagger were indeed the main issue, we might expect more 'Missing' and/or 'Mismatch' errors than 'Extra' tags. Future research could investigate the influence of error propagation in more detail.

Feature Type	CAMR Abbreviation
Left sibling unigram	*s0_slsb_ne, s0_slsb_w, s0_slsb_lemma, s0_slsb_t, s0_slsb_dl*
Right sibling unigram	*s0_srsb_ne, s0_srsb_w, s0_srsb_lemma, s0_srsb_t, s0_srsb_dl*
Second right sibling unigram	*s0_sr2sb_ne, s0_sr2sb_w, s0_sr2sb_lemma, s0_sr2sb_t, s0_sr2sb_dl*
Left sibling bigram	*s0_slsb_lemma+b0_t, s0_slsb_lemma+b0_dl, s0_slsb_t+b0_lemma, s0_slsb_dl+b0_lemma, s0_slsb_ne+b0_ne, a0_cpt+b0_cpt+s0_slsb_lemma*
Right sibling bigram	*s0_srsb_lemma+b0_dl, s0_srsb_lemma+b0_t, s0_srsb_t+b0_lemma, s0_srsb_dl+b0_lemma, s0_srsb_ne+b0_ne, a0_cpt+b0_cpt+s0_srsb_lemma*
Second right sibling bigram	*s0_sr2sb_lemma+b0_t, s0_sr2sb_lemma+b0_dl, s0_sr2sb_t+b0_lemma, s0_sr2sb_dl+b0_lemma, s0_sr2sb_ne+b0_ne, a0_cpt+b0_cpt+s0_sr2sb_lemma*

Table 5: Features added to the CAMR parser targeting bigram features of dependency tree siblings

7 Conclusions and Future Directions

In this project, I attempted various strategies for improving NE tagging in Chinese AMR using the CAMR parser. Overall, the approach with the best performance was to normalize NE tags in a preprocessing step, and then training the CAMR parser model with baseline features plus unigram and bigram features pertaining to siblings in the dependency tree.

My approach to NE normalization was preliminary, and likely contained some errors and inconsistencies. Future work could examine NE normalization in greater detail, for Chinese, English, and

Approach	Extra	Missing	Mismatch	Match	P	R	F1
Basic	198	167	73	124	.59	.56	.57
Basic+postproc	198	167	72	125	-	-	-
Sibling unigram	191	173	65	129	.60	.54	.57
Sibling unigram+postproc	191	173	63	131	-	-	-
Sibling bigram	195	160	74	131	.59	.56	.57
Sibling bigram+postproc	195	160	74	131	-	-	-
Preproc+retrained	184	164	72	133	.62	.56	.59

Table 6: Results of various Chinese AMR NE approaches

other languages. It would also be interesting to perform an inter-annotator agreement study for AMR named entities.

References

Cai, Shu & Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 2: Short papers)*, vol. 2, 748–752.

Damonte, Marco, Shay B Cohen & Giorgio Satta. 2016. An incremental parser for abstract meaning representation. *arXiv preprint arXiv:1608.06111* .

Manning, Christopher, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard & David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 55–60.

Wang, Chuan, Sameer Pradhan, Xiaoman Pan, Heng Ji & Nianwen Xue. 2016. Camr at semeval-2016 task 8: An extended transition-based amr parser. In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, 1173–1178.

Wang, Chuan, Nianwen Xue & Sameer Pradhan. 2015. A transition-based algorithm for amr parsing. In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 366–375.

Wang, Zhiguo & Nianwen Xue. 2014. Joint pos tagging and transition-based constituent parsing in chinese with non-local features. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, vol. 1, 733–742.

Xue, Naiwen, Fei Xia, Fu-Dong Chiou & Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering* 11(2). 207–238.