Human Language Technology Research Institute

Fine-Grained Opinion Extraction with Markov Logic Networks

Luis Gerardo Mojica and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas

Fine-Grained Opinion Extraction

- Involves extracting opinions from text documents
- Different from document-level opinion mining
  - E.g., determine whether a review is thumbs up or thumbs down
- Occurs at the sentence and phrase levels

Fine-Grained Opinion Extraction

- Subtask 1: Entity extraction
  - Extracts three types of entities
    - **opinions**
    - their **sources** (**who** expressed the opinions?)
    - their **targets** (what the opinions are **about**)

Fine-Grained Opinion Extraction

- Subtask 1: Entity extraction
  - Extracts three types of entities
    - **opinions**
    - their **sources** (**who** expressed the opinions?)
    - their **targets** (what the opinions are **about**)

The agency considered that the trade was favorable, but their partners are still not satisfied.

Fine-Grained Opinion Extraction

- Subtask 1: Entity extraction
  - Extracts three types of entities
    - **opinions**
    - their **sources** (**who** expressed the opinions?)
    - their **targets** (what the opinions are **about**)

> The agency considered that the trade was favorable,
>
> but their partners are still not satisfied.

Fine-Grained Opinion Extraction

- Subtask 1: Entity extraction
  - Extracts three types of entities
    - **opinions**
    - their **sources** (**who** expressed the opinions?)
    - their **targets** (what the opinions are **about**)

The agency considered that the trade was favorable, but their partners are still not satisfied.

Fine-Grained Opinion Extraction

- Subtask 1: **Entity extraction**
  - Extracts three types of entities
    - **opinions**
    - their **sources** (**who** expressed the opinions?)
    - their **targets** (what the opinions are **about**)

The agency considered that the trade was favorable, but their partners are still not satisfied.

Fine-Grained Opinion Extraction

- Subtask 2: **Relation extraction**
  - Extracts two types of relations
    - **is_from** (between an opinion and its source)
    - **is_about** (between an opinion and its target)

> The agency considered that the trade was favorable,
>
> but their partners are still not satisfied.

Fine-Grained Opinion Extraction

- Subtask 2: **Relation extraction**
  - Extracts two types of relations
    - **is_from** (between an opinion and its source)
    - **is_about** (between an opinion and its target)

is_from

The agency considered that the trade was favorable,

but their partners are still not satisfied.

Fine-Grained Opinion Extraction

- Subtask 2: **Relation extraction**
  - Extracts two types of relations
    - **is_from** (between an opinion and its source)
    - **is_about** (between an opinion and its target)

The agency considered that the trade was favorable, but their partners are still not satisfied.

is_from

Fine-Grained Opinion Extraction

- Subtask 2: **Relation extraction**
  - Extracts two types of relations
    - **is_from** (between an opinion and its source)
    - **is_about** (between an opinion and its target)

is_about

The agency considered that the trade was favorable,

but their partners are still not satisfied.

Fine-Grained Opinion Extraction

- Subtask 2: **Relation extraction**
  - Extracts two types of relations
    - **is_from** (between an opinion and its source)
    - **is_about** (between an opinion and its target)

The agency considered that the trade was favorable,

but their partners are still not satisfied.

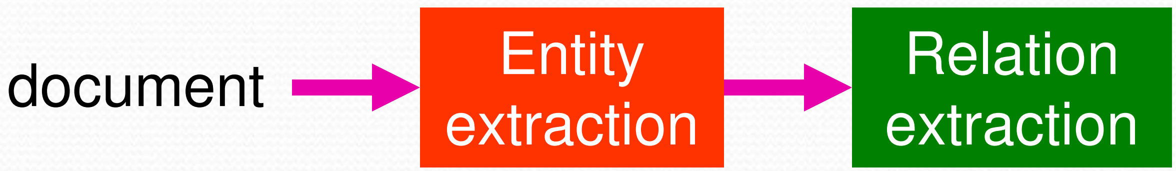
is_about

Challenges

- Two opinions can share the same target
  - the trade is the target of both considered and not satisfied
- An opinion can be associated with more than one source/target
- Whether a word is an opinion is context-dependent
  - a given word can sometimes be an opinion and sometimes not

Previous Approaches

- **Pipeline** approach



Extract the 3 types of entities

For each pair of entities extracted, determine what type of relation exists between them, if any

Weakness of the Pipeline Approach

- **Error propagation**
  - Errors made by the entity extraction component will be propagated to the relation extraction component

The agency considered that the trade was favorable,

but their partners are still not satisfied.

Weakness of the Pipeline Approach

- **Error propagation**
  - Errors made by the entity extraction component will be propagated to the relation extraction component

The agency considered that the trade was favorable,

but their partners are still not satisfied.

Weakness of the Pipeline Approach

- **Error propagation**
  - Errors made by the entity extraction component will be propagated to the relation extraction component

The agency considered that the trade was favorable,

but their partners are still not satisfied.

is_about

Addressing Error Propagation

- **Integer Linear Programming** (ILP) [Yang & Cardie, 2013]
  - To be robust to the errors, generate lots of entity candidates

The agency considered that the trade was favorable, but their partners are still not satisfied.

Addressing Error Propagation

- **Integer Linear Programming** (ILP) [Yang & Cardie, 2013]
  - To be robust to the errors, generate lots of entity candidates

0.9 0.8 0.2 0.2

The agency considered that the trade was favorable,

but their partners are still not satisfied.

0.7 0.9

Addressing Error Propagation

- **Integer Linear Programming** (ILP) [Yang & Cardie, 2013]
  - To be robust to the errors, generate lots of entity candidates

0.9 0.8 0.2 0.2

The agency considered that the trade was favorable,

but their partners are still not satisfied.

0.7 0.9

Addressing Error Propagation

- **Integer Linear Programming** (ILP) [Yang & Cardie, 2013]
  - To be robust to the errors, generate lots of entity candidates

0.9 0.8 0.2 0.2

The agency considered that the trade was favorable,
but their partners are still not satisfied.

0.7 0.9

Addressing Error Propagation

- **Integer Linear Programming** (ILP) [Yang & Cardie, 2013]
  - To be robust to the errors, generate lots of entity candidates

0.9 0.8 0.2 0.2

The agency considered that the trade was favorable,

but their partners are still not satisfied.

0.7 0.9

Addressing Error Propagation

- **Integer Linear Programming** (ILP) [Yang & Cardie, 2013]
  - To be robust to the errors, generate lots of entity candidates

0.9 0.8 0.2 0.2

The agency considered that the trade was favorable,

but their partners are still not satisfied.

0.7 0.9

is_about 0.9

Addressing Error Propagation

- **Integer Linear Programming** (ILP) [Yang & Cardie, 2013]
  - To be robust to the errors, generate lots of entity candidates

0.9 0.8 0.2 0.2

The agency considered that the trade was favorable,

but their partners are still not satisfied.

0.7 0.9

is_about 0.9

Integer Linear Programming (ILP)

- A constrained optimization framework
  - Optimize an objective function subject to linear constraints

- For fine-grained opinion extraction,
  - **Objective function**: combines confidence values from the classifiers trained for both subtasks
  - **Goal**: re-classify each test instance so that the resulting set of classifications collectively optimize the objective function

Integer Linear Programming (ILP)

- A constrained optimization framework
  - Optimize an objective function subject to linear constraints
- For fine-grained opinion extraction,
  - **Objective function**: combines confidence values from the classifiers trained for both subtasks
  - **Goal**: re-classify each test instance so that the resulting set of classifications collectively optimize the objective function
- This is a **joint inference** process
  - When optimizing objective function, test instances from the subtasks are not being re-classified **independently**
    - Both subtasks can influence each other

Constraints for ILP

- The constraints are important
  - Constraints we want the outputs of the 2 subtasks to satisfy
  - E.g., if two entity candidates have an is_from relation, then one of them has to be a source and the other has to be an opinion

- Designing good constraints is crucial to ILP's performance

Our Goal

- Improve the state of the art on this task by proposing
  - **New feature:** feature derived from a factuality lexicon

Our Goal

- Improve the state of the art on this task by proposing
  - **New feature:** feature derived from a factuality lexicon
  - **New approach:** Markov Logic Networks (MLNs)
    - can perform joint inference
    - but much less used in NLP tasks than ILP

MLNs: Better than ILP?

- MLNs allow constraints to be specified in a more intuitive and compact manner
  - ILP is propositional, MLNs employ first-order logic

MLNs: Better than ILP?

- MLNs allow constraints to be specified in a more intuitive and compact manner
  - ILP is propositional, MLNs employ first-order logic
- MLNs make it easy to specify **soft** constraints
  - not easy to encode soft constraints in ILP

Plan for the Talk

- Corpus
- Baseline systems
- Our approach
- Evaluation

Plan for the Talk

- Corpus
- Baseline systems
- Our approach
- Evaluation

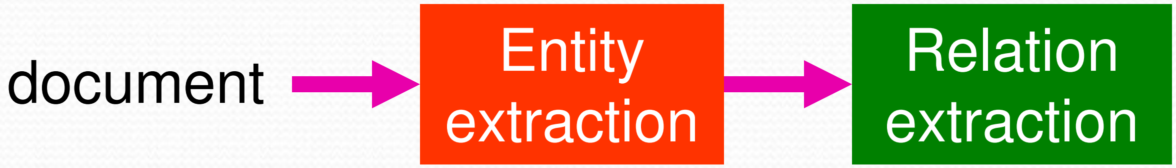
Corpus

- MPQA 2.0 corpus
  - 433 documents
    - 8377 sentences
      - 4717 opinions, 4680 targets, and 5505 sources
      - 13046 is_about relations, 9763 is_from relations

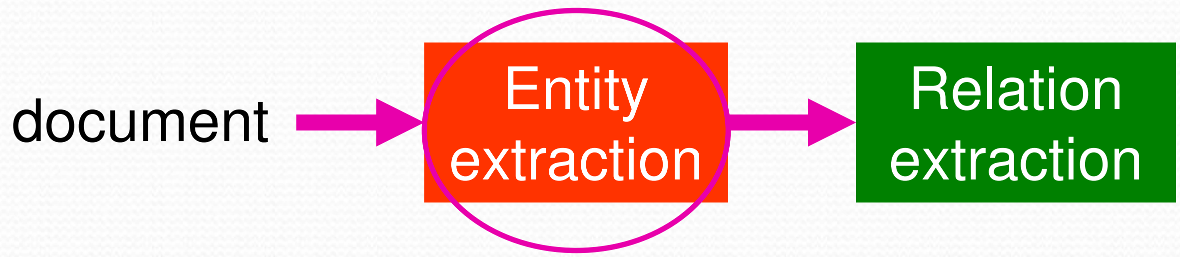
Plan for the Talk

- Corpus
- Baseline systems
- Our approach
- Evaluation

Baseline 1: Pipeline Approach

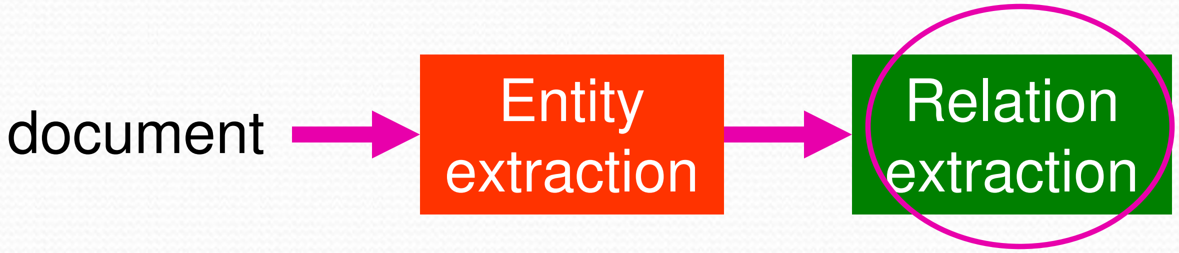


Baseline 1: Pipeline Approach



- To train the entity extraction model,
  - Recast the task as a sequence labeling task
    - Each training instance corresponds to a word token
      - 4 types of features
    - Trained a CRF model

Baseline 1: Pipeline Approach



- For relation extraction,
  - Train two binary SVM classifiers (is_from and is_about)
    - To create training instances for these classifiers,
      - pair each opinion with each source/target
      - 2 types of features
  - A test instance is created by pairing each opinion with each source/target extracted by the CRF

Baseline 2: Yang & Cardie's ILP Approach

- ILP: a constrained optimization framework
  - **Goal**: optimize objective function (composed of the confidence values returned by the CRF and the SVM classifiers) subject to a set of linear constraints
    - constraints taken from Y&C
- Need to generate many entity candidates
  - Obtain them the 30-best CRF outputs

Plan for the Talk

- Corpus
- Baseline systems
  - Pipeline approach
  - Yang & Cardie's ILP approach
- Our approach
  - New feature based on factuality lexicon
  - MLN formulation
- Evaluation

Factuality Lexicon

Mary suspects that John left Miami.

Mary knows that John left Miami.

Factuality Lexicon

Mary suspects that John left Miami.

Mary knows that John left Miami.

- Sauri (2009) divided verbs into 49 categories
  - suspects belongs to category Conjecture
  - knows belongs to category Disclose

Factuality Lexicon

Mary suspects that John left Miami.

Mary knows that John left Miami.

- Sauri (2009) divided verbs into 49 categories
  - suspects belongs to category Conjecture
    - verbs in Conjecture are likely to correspond to opinions
  - knows belongs to category Disclose
    - verbs in Disclose are likely to correspond to facts

Factuality Lexicon

Mary suspects that John left Miami.

Mary knows that John left Miami.

- Sauri (2009) divided verbs into 49 categories
  - suspects belongs to category Conjecture
    - verbs in Conjecture are likely to correspond to opinions
  - knows belongs to category Disclose
    - verbs in Disclose are likely to correspond to facts
  - These categories are helpful for identifying opinions
  - Train the CRF with an **additional feature**
    - value is the category to which the verb belongs

MLN Formulation: OpinMLN

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

MLN Formulation: Opin

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

4 predicates

Query predicates:
Type(i,c)
Is_about(i,j)
Is_from(i,j)

Evidence predicates:
Overlap(i,j)
OneBest(i,c)

MLN Formulation: Opin

4 predicates

Query predicates:
Type(i,c)
Is_about(i,j)
Is_from(i,j)

Evidence predicates:
Overlap(i,j)
OneBest(i,c)

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

MLN Formulation: Opin

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

4 predicates

Query predicates:
Type(i,c)
Is_about(i,j)
Is_from(i,j)

Evidence predicates:
Overlap(i,j)
OneBest(i,c)

MLN Formulation: Opin

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

4 predicates

Query predicates:
Type(i,c)
Is_about(i,j)
Is_from(i,j)

Evidence predicates:
Overlap(i,j)
OneBest(i,c)

MLN Formulation: OpinMLN

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

A span i cannot have any relation with itself

MLN Formulation: OpinMLN

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

If the 1-best CRF output says span i has entity type c, we will label span i as an entity with type c

MLN Formulation: OpinMLN

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

First 3 are to be enforced as **hard constraints**

MLN Formulation: OpinMLN

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

The remaining constraints are to be enforced as **soft constraints**

Weight indicates how important it is to satisfy the constraint

MLN Formulation: OpinMLN

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

If span i is in an is_from relation with span j, then i should be an opinion and j should be a source

MLN Formulation: OpinMLN

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Ty

If span i is in an is_about relation with span j, then i should be an opinion and j should be a target

MLN Formulation: OpinMLN

1) !Is_about(i,i) .
2) !Is_from(i,i) .
3) OneBest(i,c) → Type(i,c) .
4) w_4 Is_from(i,j) → Type(i,O)
5) w_5 Is_from(i,j) → Type(j,S)
6) w_6 Is_about(i,j) → Type(i,O)
7) w_7 Is_about(i,j) → Type(j,T)
8) w_8 Overlap(i,j) → Type(i,N) v Type(j,N)

If span i overlaps with span j, then either i or j is not a real entity

Incorporating Prior Knowledge

- Like ILP, the MLN exploits the CRF and SVM's outputs
  - Model their outputs as **soft evidence**
    - Our prior belief that a grounded query predicate is true

Plan for the Talk

- Corpus
- Baseline systems
  - Pipeline approach
  - Yang & Cardie's ILP approach
- Our approach
  - New feature based on factuality lexicon
  - MLN formulation
- Evaluation

Evaluation

- MPQA 2.0 corpus
  - 433 documents
    - 397 documents for training, 36 documents for testing
- Evaluation metrics
  - precision, recall, F1-score for both subtasks

Results: Entity Extraction

	Opinion F1	Target F1	Source F1
Pipeline	54.9	38.5	59.3
Duplicated Y&C's ILP	59.4	40.1	48.1

Results: Entity Extraction

	Opinion F1	Target F1	Source F1
Pipeline	54.9	38.5	59.3
Duplicated Y&C's ILP	59.4	40.1	48.1

- ILP is better than Pipeline on Opinion and Target extraction but worse on Source extraction
  - ILP doesn't always yield improvements

Results: Entity Extraction

	Opinion F1	Target F1	Source F1
Pipeline	54.9	38.5	59.3
Duplicated Y&C's ILP	59.4	40.1	48.1
OpinMLN+factuality	59.1	43.5	62.1

- Our MLN approach performs significantly better than the two baselines on Source and Target extraction
  - Statistically tied with ILP on Opinion extraction

Results: Relation Extraction

	is_from F1	is_about F1
Pipeline	19.8	22.7
Duplicated Y&C's ILP	14.2	19.4

- ILP underperforms Pipeline

Results: Relation Extraction

	is_from F1	is_about F1
Pipeline	19.8	22.7
Duplicated Y&C's ILP	14.2	19.4
OpinMLN+factuality	21.4	32.4

- Our MLN approach outperforms both baselines significantly on both relation types

Summary

- presented the first MLN formulation for fine-grained opinion extraction
- showed that OpinMLN significantly outperformed Y&C's state-of-the-art ILP approach on the MPQA corpus when used in combination with factuality