**Person/ORG – Location – Time**

**Data Sources :** News articles from Internet in 3 languages namely English, Bengali and German.

**Objective:** Associate a person-location-time(of event) or organization-location-time(of event) triplet for each news article in such a way that this triplet will describe about most important person or organization, time and location within corresponding news article. The reason the person/organization is at a particular place in a particular time will be clear by examining/reading /understanding live link to source article by human.

**Corpora Collection:**

1. ***News Articles:*** Using full name and partial name( first/last names) of selected persons manually news articles are collected in 3 languages(English, Bengali and German) from different news sources searching through Google search engine from Bangladesh (search performance varies with location according to Google search engine phenomenon).

**Association Rules Used:**

Supervised training will try to find some pattern that will reveal most important PLT or OLT triplet. Possible PLT/OLT (triplet) formation rules are given below:

|  |
| --- |
| 1. Make combination of all possible pairs of location and time within articles. With every person/organization associate every pairs {location - time} formed above to form possible triplets PLT/OLT. |
| 1. There must be at least one person/organization. Sources are collected in such a way. |
| 1. There may be no suitable time or location within an article. |
| 1. If no suitable location or time is found then take “-1” to describe this situation. See Guess tag description in c) and d) subsection. |
| 1. Elements from {person/organization}set mapped to elements of{location}set will be an onto function. That means every location should be associated with at least one person/organization. |
| 1. Elements from {person/organization}set mapped to elements of {time}set will be an onto function. That means every time should be associated with at least one person/organization. |
| 1. The time is specific indication of a day or month or year e.g. “Monday” or “March 1”, “2014” or “March” or “2014” etc. |
| 1. Locations are name of a place or an indication of a place. |
| 1. Name of a country will be a location if it actually indicates a particular country as a location/land/place. If name of a country/city indicates government role then it will not be a location, it’ll be an organization entity. |

Annotation descriptions are listed below:

|  |  |
| --- | --- |
| TAG | EXPLANATION |
| Main | Contains all other tags within it and its the first tag of every file. |
| source | Source: Internet source address of corresponding file. |
| language | Reporting language of the file. |
| answer | Contains information for supervised traning processes. Its attributes are:   1. “accuracy” contains value for “PLTclassifier”. “1” = guessed (P/O)LT combination is correct,   “0” = guessed (P/O)LT combination is not correct.   1. “lebel\_PL” contains value for “PLclassifier”. “1” = guessed PL combination is correct,   “0” = guessed PL combination is not correct.   1. “lebel\_PT” contains value for “PTclassifier”. “1” = guessed PT combination is correct,   “0” = guessed PT combination is not correct.   1. “lebel\_OL” contains value for “OLclassifier”. “1” = guessed OL combination is correct,   “0” = guessed OL combination is not correct.   1. “lebel\_OT” contains value for “OTclassifier”. “1” = guessed OT combination is correct,   “0” = guessed OT combination is not correct. |
| Guess | Expresses which combination of (P/O)LT is currently considering. Its attributes are:   1. “PersonID” , takes value from this set {“-1”, “1”, “2”, .... any positive integer}. “-1” means alternative organization is considering for now. Other positive numbers are for different person ID representing their unique identity within the article. 2. “OrganizationID” , takes value from this set {“-1”, “1”, “2”, .... any positive integer}. “-1” means alternative person is considering for now. Other positive numbers are for different organization representing their unique identity within the article. 3. “TimeID” , takes value from this set {“-1”, “1”, “2”, .... any positive integer}. “-1” means no suitable time is present to consider. Other positive numbers are for different times representing their unique identity within the article. 4. “LocationID” , takes value from this set {“-1”, “1”, “2”, .... any positive integer}. “-1” means no suitable location is present to consider. Other positive numbers are for different locations representing their unique identity within the article.   Note: At least one of two tags namely from set { PersonID , OrganizationID } will take value “-1” but not both simultaneously . Values of {a,b,c} or {b,c,d} represents current considering combination for PLT or OLT. If a = “-1” then {b,c,d} is considered. If b = “-1” then {a,c,d} is considered. |
| description | Contains description of all entities within corresponding article with their corresponding attribute values. Entities are described using tags: “PERSON”, for person entity; “Organization”, for organization entity; “TIME”, for time entity; “LOCATION”, for location entity. |
| Title | Contains title description of news article. |
| text | Contains main body of the news articles collected from “source” tag address above. Here articles are in normalized form meaning articles are manually reduced to some arbitrary suitable size maintaining actual association between sentences that was in main web source text, starting from first sentence of the text in web. |
| s | “Contains a single sentence” or “Strings of characters that can be considered as a single sentence”. |
| speaker | Tries to answer who is talking within current considering sentence, e.g. speaker of a quote. |
| PERSON | Uniquely identifies every person entity within article. Its “PersonID” attribute is described above under Guess tag description in a) subsection. Co-resolution is resolved for pronoun. |
| Organization | Uniquely identifies every organization entity within article. Its “OrganizationID” attribute is described above under Guess tag description in b) subsection. Co-resolution is resolved for pronoun. |
| TIME | Uniquely identifies every time entity within article. Its “TimeID” attribute is described above under Guess tag description in c) subsection. Co-resolution is resolved for pronoun. |
| LOCATION | Uniquely identifies every location entity within article. Its “LocationID” attribute is described above under Guess tag description in d) subsection. Co-resolution is resolved for pronoun. |
| n | Nothing to do with this tag.(if exist) |

*No of files within data folder.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Jolie | Obama | Putin | Gates | Hilary |
| English | 10 | 10 | 10 | 10 | 10 |
| German | 10 | 10 | 10 | 10 | 10 |
| Bengali | 10 | 10 | 10 | 10 | 10 |

**Assumption:** If any SVM successfully classify we’ll say it class “1”, else class “0”. For extracting related patterns for PLT/OLT we follow architecture below:

1. Train PL classifier (SVM) using features listed below section A.
2. Train PT classifier (SVM) using features listed below section B.
3. Train OL classifier (SVM) using features listed below section C.
4. Train OT classifier (SVM) using features listed below section D.
5. Train PLT classifier (SVM) using features listed below section E (Ensemble).

Based on “answer” and “Guess” tags in xml we’ll take decision for 1,2,3 and 4 steps above. For example, If “accuaracy” = “1” and “personID” = “-1” then OL classifier and OT classifier both should return “1”, If “accuaracy” = “0” and “personID” = “-1” then OL classifier and OT classifier both should return “0” and for above mentioned any situation PL classifier and PT classifier both should return “0”.

We can derive PLT classifier value based on others classifiers value.

1. If PL classifier and PT classifier classifies successfully then PLT classifier will successfully classify. Returns “1” else “0”.
2. If OL classifier and OT classifier classifies successfully then PLT classifier will successfully classify. Returns “1” else “0”.
3. If (1) or (2) returns “1” then PLT classifier returns “1” for some predefined triplet. In every other situation PLT classifier returns “0”.

**Feature Description:**

1. **For Per-Loc:**

|  |  |
| --- | --- |
| Features |  |
| 1. double 🡨 quotation   Rule:  double quotation()  {  if( speaker\_name == person\_name )  {  np = Count(person\_name);// within quote  nl = Count(location\_name);    not\_np = Count(!person\_name);  not\_nl = Count(!location\_name);  return ( 5.0\*( np+nl ) - 2.0\*( not\_nl + not\_np ) );    }  else{    np = Count(person\_name);// within quote  nl = Count(location\_name);    not\_np = Count(!person\_name);  not\_nl = Count(!location\_name);  return ( ( np+nl ) - ( not\_nl + not\_np ) );    }  } |  |
| 1. Count(person\_name) // within whole art. |  |
| 1. Count(location\_name) // within whole art. |  |
| 1. double 🡨 CoExistence.   double CoExistence  {  double cc = 0.0;  for( sentenceNo = 1 to last )  {  cc += Count\_CoExistence\_InSentence( sentenceNo , person\_name, location\_name);  }  return cc;  } |  |
| 1. if( MatchExist( titled\_location, location\_name)) return 5.0;   else return 0.0; |  |
| 1. if( MatchExist( titled\_person, person \_name)) return 5.0;   else return 0.0; |  |
| 1. entConcentration ()   {  // for person\_name  double nfo = 0.0;  for( sentenceNo = 1 to last )  {  nfo += Count(NoOfSentenceBefore\_NextOccurance);  }  return ( nfo / last )\*( -1.0);  } |  |
| 1. double entConcentration()   {  // for location \_name  double nfo = 0.0;  for( sentenceNo = 1 to last )  {  nfo += Count(NoOfSentenceBefore\_NextOccurance);  }  return ( nfo / last )\*( -1.0);  } |  |
| 1. summation of ent-concentrations( both ) in first 2 paragraph/ first 8 lines |  |

1. **For Per-Time:**

|  |  |
| --- | --- |
| Features | Comment |
| 1. double 🡨 quotation   Rule:  double quotation()  {  if( speaker\_name == person\_name )  {  np = Count(person\_name);// within quote  nt = Count(time);    not\_np = Count(!person\_name);  not\_nt = Count(!time);  return ( 5.0\*( np+nt ) - 2.0\*( not\_nt + not\_np ) );    }  else{    np = Count(person\_name);// within quote  nt = Count(time);    not\_np = Count(!person\_name);  not\_nt = Count(!time);  return ( ( np+nt ) - ( not\_nt + not\_np ) );    }  } |  |
| 1. Count(person\_name) // within whole art. |  |
| 1. Count(time) // within whole art. |  |
| 1. double 🡨 CoExistence.   double CoExistence  {  double cc = 0.0;  for( sentenceNo = 1 to last )  {  cc += Count\_CoExistence\_InSentence( sentenceNo , person\_name, time);  }  return cc;  } |  |
| 1. if( MatchExist( titled\_ time, time)) return 5.0;   else return 0.0; |  |
| 1. if( MatchExist( titled\_person, person \_name)) return 5.0;   else return 0.0; |  |
| 1. entConcentration ()   {  // for person\_name  double nfo = 0.0;  for( sentenceNo = 1 to last )  {  nfo += Count(NoOfSentenceBefore\_NextOccurance);  }  return ( nfo / last )\*( -1.0);  } |  |
| 1. double entConcentration()   {  // for time  double nfo = 0.0;  for( sentenceNo = 1 to last )  {  nfo += Count(NoOfSentenceBefore\_NextOccurance);  }  return ( nfo / last )\*( -1.0);  } |  |
| 1. summation of ent-concentrations( both ) in first 2 paragraph/ first 8 lines |  |

1. **For ORG-Loc:**

|  |  |
| --- | --- |
| Features | Comment |
| 1. double 🡨 quotation   Rule:  double quotation()  {  if( speaker\_name == ORG \_name )  {  np = Count(ORG\_name);// within quote  nl = Count(location\_name);    not\_np = Count(!ORG\_name);  not\_nl = Count(!location\_name);  return ( 5.0\*( np+nl ) - 2.0\*( not\_nl + not\_np ) );    }  else{    np = Count(ORG\_name);// within quote  nl = Count(location\_name);    not\_np = Count(!ORG\_name);  not\_nl = Count(!location\_name);  return ( ( np+nl ) - ( not\_nl + not\_np ) );    }  } |  |
| 1. Count(ORG\_name) // within whole art. |  |
| 1. Count(location\_name) // within whole art. |  |
| 1. double 🡨 CoExistence.   double CoExistence  {  double cc = 0.0;  for( sentenceNo = 1 to last )  {  cc += Count\_CoExistence\_InSentence( sentenceNo , ORG\_name, location\_name);  }  return cc;  } |  |
| 1. if( MatchExist( titled\_location, location\_name)) return 5.0;   else return 0.0; |  |
| 1. if( MatchExist( titled\_ORG, ORG \_name)) return 5.0;   else return 0.0; |  |
| 1. entConcentration ()   {  // for ORG\_name  double nfo = 0.0;  for( sentenceNo = 1 to last )  {  nfo += Count(NoOfSentenceBefore\_NextOccurance);  }  return ( nfo / last )\*( -1.0);  } |  |
| 1. double entConcentration()   {  // for location \_name  double nfo = 0.0;  for( sentenceNo = 1 to last )  {  nfo += Count(NoOfSentenceBefore\_NextOccurance);  }  return ( nfo / last )\*( -1.0);  } |  |
| 1. summation of ent-concentrations( both ) in first 2 paragraph/ first 8 lines |  |

1. **For ORG-Time:**

|  |  |
| --- | --- |
| Features | Comment |
| 1. double 🡨 quotation   Rule:  double quotation()  {  if( speaker\_name == ORG\_name )  {  np = Count(ORG\_name);// within quote  nt = Count(time);    not\_np = Count(!ORG\_name);  not\_nt = Count(!time);  return ( 5.0\*( np+nt ) - 2.0\*( not\_nt + not\_np ) );    }  else{    np = Count(ORG\_name);// within quote  nt = Count(time);    not\_np = Count(!ORG\_name);  not\_nt = Count(!time);  return ( ( np+nt ) - ( not\_nt + not\_np ) );    }  } |  |
| 1. Count(ORG\_name) // within whole art. |  |
| 1. Count(time) // within whole art. |  |
| 1. double 🡨 CoExistence.   double CoExistence  {  double cc = 0.0;  for( sentenceNo = 1 to last )  {  cc += Count\_CoExistence\_InSentence( sentenceNo , ORG\_name, time);  }  return cc;  } |  |
| 1. if( MatchExist( titled\_ time, time)) return 5.0;   else return 0.0; |  |
| 1. if( MatchExist( titled\_ORG, ORG \_name)) return 5.0;   else return 0.0; |  |
| 1. entConcentration ()   {  // for ORG\_name  double nfo = 0.0;  for( sentenceNo = 1 to last )  {  nfo += Count(NoOfSentenceBefore\_NextOccurance);  }  return ( nfo / last )\*( -1.0);  } |  |
| 1. double entConcentration()   {  // for time  double nfo = 0.0;  for( sentenceNo = 1 to last )  {  nfo += Count(NoOfSentenceBefore\_NextOccurance);  }  return ( nfo / last )\*( -1.0);  } |  |
| 1. summation of ent-concentrations( both ) in first 2 paragraph/ first 8 lines |  |

1. **PER/ORG – LOC – TIME:**

|  |  |
| --- | --- |
| Features | Comment |
| 1. PL classifier classification value “0” or “1” |  |
| 1. PT classifier classification value “0” or “1” |  |
| 1. OL classifier classification value “0” or “1” |  |
| 1. OT classifier classification value “0” or “1” |  |
| 1. CoExistence(P-L) + CoExistence( P-T) |  |
| 1. CoExistence(P-L-T) |  |
| 1. CoExistence( L-T ) |  |