Multichannel decoding of multiparty meetings.

Joe Frankel

CSTR Meeting, 21st November 2006

Talk outline

Background

- dialogue acts
- motivation
- baseline
- DBN decoding

Single-channel DBN

- structure
- results
- adding activity information

multichannel DBN

- structure
- results

Future work

Dialogue acts (DA)

Dialogue acts (DA) are the fundamental building blocks of a conversation.

Dialogue acts (DA)

Dialogue acts (DA) are the fundamental building blocks of a conversation.

- Set of 5 dialogue acts used in this work:

Statements	yeah that's right. maybe
Questions	right? why twenty eight
Backchannels	yeah right
Floorgrabbers	um- so- uh uh-
Disruptions	but- and it doesn't-

Dialogue acts (DA)

Dialogue acts (DA) are the fundamental building blocks of a conversation.

- Set of 5 dialogue acts used in this work:

Statements	yeah that's right. maybe
Questions	right? why twenty eight
Backchannels	yeah right
Floorgrabbers	um- so- uh uh-
Disruptions	but- and it doesn't-

- Dialogue act recognition:

Dialogue acts (DA)

Dialogue acts (DA) are the fundamental building blocks of a conversation.

- Set of 5 dialogue acts used in this work:

Statements	yeah that's right. maybe
Questions	right? why twenty eight
Backchannels	yeah right
Floorgrabbers	um- so- uh uh-
Disruptions	but- and it doesn't-

- Dialogue act recognition:
  - given the transcript of a meeting, determine the sequence of dialogue acts for each participant

Motivation 1

This work came about after talking to Matthias Zimmermann at MLMI 2006.

Motivation 1

This work came about after talking to Matthias Zimmermann at MLMI 2006.

- His work involved the application of a number of machine learning techniques to DA recognition. These include:

Motivation 1

This work came about after talking to Matthias Zimmermann at MLMI 2006.

- His work involved the application of a number of machine learning techniques to DA recognition. These include:
  - Hidden-Event Language models (LM)

Motivation 1

This work came about after talking to Matthias Zimmermann at MLMI 2006.

- His work involved the application of a number of machine learning techniques to DA recognition. These include:
  - Hidden-Event Language models (LM)
  - Maximum Entropy

Motivation 1

This work came about after talking to Matthias Zimmermann at MLMI 2006.

- His work involved the application of a number of machine learning techniques to DA recognition. These include:
  - Hidden-Event Language models (LM)
  - Maximum Entropy
  - Conditional Random Fields (CRF)

Motivation 1

This work came about after talking to Matthias Zimmermann at MLMI 2006.

- His work involved the application of a number of machine learning techniques to DA recognition. These include:
  - Hidden-Event Language models (LM)
  - Maximum Entropy
  - Conditional Random Fields (CRF)
- The models were trained to identify words which conclude dialogue acts, i.e. does this word represent a DA boundary, and if so, which type?.

Motivation 1

This work came about after talking to Matthias Zimmermann at MLMI 2006.

- His work involved the application of a number of machine learning techniques to DA recognition. These include:
  - Hidden-Event Language models (LM)
  - Maximum Entropy
  - Conditional Random Fields (CRF)
- The models were trained to identify words which conclude dialogue acts, i.e. does this word represent a DA boundary, and if so, which type?.
  - The same models could therefore be applied for segmentation and classification, though decoding was implemented as a two-stage segmentation then classification.

Motivation 2

I became interested in building a probabilistic framework in which to:

Motivation 2

I became interested in building a probabilistic framework in which to:

- perform a full Viterbi decoding of dialogue act states (rather than segmentation followed by classification).

Motivation 2

I became interested in building a probabilistic framework in which to:

- perform a full Viterbi decoding of dialogue act states (rather than segmentation followed by classification).
- jointly decode all channels (participants) together, rather than independently of one-another

Motivation 2

I became interested in building a probabilistic framework in which to:

- perform a full Viterbi decoding of dialogue act states (rather than segmentation followed by classification).
- jointly decode all channels (participants) together, rather than independently of one-another
  - previous approaches have typically considered one channel at a time, ignoring information on the overall state of the system.

Admission

This talk is not really about dialogue act decoding!

It's about a way of quickly building a general decoder. It may work for you if you are working with temporal data, and

Admission

This talk is not really about dialogue act decoding!

It's about a way of quickly building a general decoder. It may work for you if you are working with temporal data, and

- your problem involves inference on multiple discrete classes

Admission

This talk is not really about dialogue act decoding!

It's about a way of quickly building a general decoder. It may work for you if you are working with temporal data, and

- your problem involves inference on multiple discrete classes
- you have a model which can produce posteriors/likelihoods for each class at each time

Admission

This talk is not really about dialogue act decoding!

It's about a way of quickly building a general decoder. It may work for you if you are working with temporal data, and

- your problem involves inference on multiple discrete classes
- you have a model which can produce posteriors/likelihoods for each class at each time
- you have multiple information sources or would like to employ multiple state streams

Admission

This talk is not really about dialogue act decoding!

It's about a way of quickly building a general decoder. It may work for you if you are working with temporal data, and

- your problem involves inference on multiple discrete classes
- you have a model which can produce posteriors/likelihoods for each class at each time
- you have multiple information sources or would like to employ multiple state streams
- you have a number of topologies/constraints which seem appropriate and you would like to try

Baseline - posteriors

Baseline results use conditional random field (CRF) posteriors.
There are two sets of posteriors, relating to:

Baseline - posteriors

Baseline results use conditional random field (CRF) posteriors. There are two sets of posteriors, relating to:

- binary decision, whether current word represents the end of a dialogue act, regardless of class

Baseline - posteriors

Baseline results use conditional random field (CRF) posteriors. There are two sets of posteriors, relating to:

- binary decision, whether current word represents the end of a dialogue act, regardless of class
- $(n+1)$-way decision relating whether the current word is internal to a dialogue act, or which of the n types it completes.

DA decoding - form of data

transcript	DA state	Posterior	
		P(final)	P(internal)
did	q	0.01	0.99
you	q	0.02	0.98
take	q	0.01	0.99
a	q	0.01	0.99
day	q	0.08	0.92
off?	**Q**	**0.95**	**0.05**
yeah	s	0.01	0.99
that's	s	0.02	0.98
right.	**S**	**0.98**	**0.02**

Baseline - decoding strategy

See Zimmermann et al. (2006) for details.

- Use binary-class posterior with threshold (tuned on development set) to decide if the current word represents the completion of a dialogue act, regardless of type.

Baseline - decoding strategy

See Zimmermann et al. (2006) for details.

- Use binary-class posterior with threshold (tuned on development set) to decide if the current word represents the completion of a dialogue act, regardless of type.
- If threshold is exceeded, use $(n+1)$-class posterior to determine the DA type.

Baseline - decoding strategy

See Zimmermann et al. (2006) for details.

- Use binary-class posterior with threshold (tuned on development set) to decide if the current word represents the completion of a dialogue act, regardless of type.
- If threshold is exceeded, use $(n+1)$-class posterior to determine the DA type.
- I present results in terms of..

Baseline - decoding strategy

See Zimmermann et al. (2006) for details.

- Use binary-class posterior with threshold (tuned on development set) to decide if the current word represents the completion of a dialogue act, regardless of type.
- If threshold is exceeded, use $(n+1)$-class posterior to determine the DA type.
- I present results in terms of..
  - strict (error): both DA and boundaries must be correct

Baseline - decoding strategy

See Zimmermann et al. (2006) for details.

- Use binary-class posterior with threshold (tuned on development set) to decide if the current word represents the completion of a dialogue act, regardless of type.
- If threshold is exceeded, use $(n+1)$-class posterior to determine the DA type.
- I present results in terms of..
  - strict (error): both DA and boundaries must be correct
  - F-measure: harmonic mean of precision and recall

Baseline - results

Experiments use the ICSI meetings corpus, with 51 meetings for training and 11 each for development and evaluation.

strict	F-measure
61.6	52.5

Approach

Approach taken is to use dynamic Bayesian networks (DBN):

Approach

Approach taken is to use dynamic Bayesian networks (DBN):

- Random variables and dependencies are encoded as nodes and edges of a directed acyclic graph (DAG).

Approach

Approach taken is to use dynamic Bayesian networks (DBN):

- Random variables and dependencies are encoded as nodes and edges of a directed acyclic graph (DAG).
- Ideal framework in which to build multi-stream models, and combine information from multiple sources.

Approach

Approach taken is to use dynamic Bayesian networks (DBN):

- Random variables and dependencies are encoded as nodes and edges of a directed acyclic graph (DAG).
- Ideal framework in which to build multi-stream models, and combine information from multiple sources.
- Inference and training algorithms are derived for entire class of DBNs, so useful for rapid prototyping of new models.

DBNs - notation and terminology

- square/round nodes denote discrete/continuous RVs

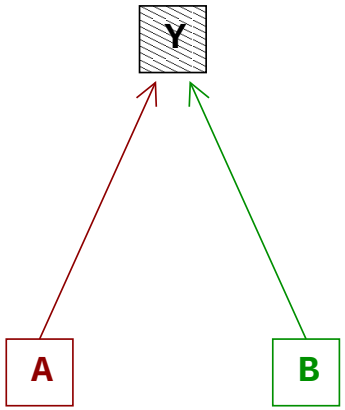
DBNs - notation and terminology

- square/round nodes denote discrete/continuous RVs
- shaded/unshaded nodes denote observed/hidden RVs

DBNs - notation and terminology

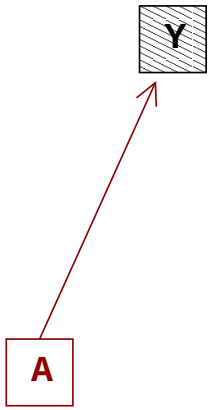
- square/round nodes denote discrete/continuous RVs
- shaded/unshaded nodes denote observed/hidden RVs
- graph is directed, so dependencies have direction. The parents of a given node are the RVs on which it is conditioned.

DBNs - factoring the joint distribution



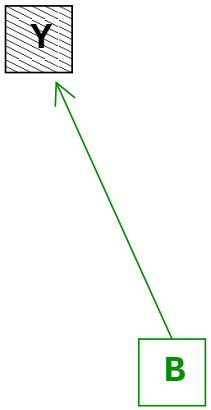
$$P(Y, A, B) = P(Y|A, B)P(A)P(B)$$

DBNs - switching parents



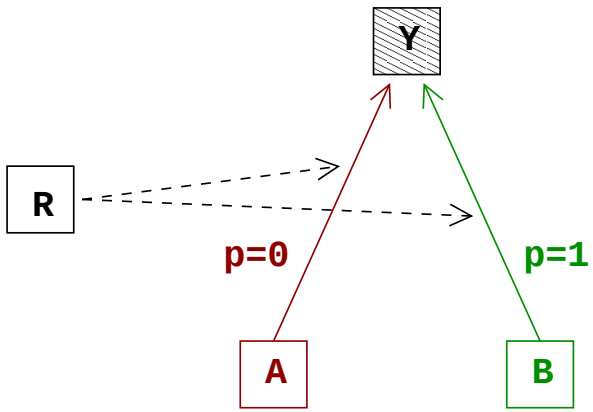
$$P(Y, A) = P(Y|A)P(A)$$

DBNs - switching parents



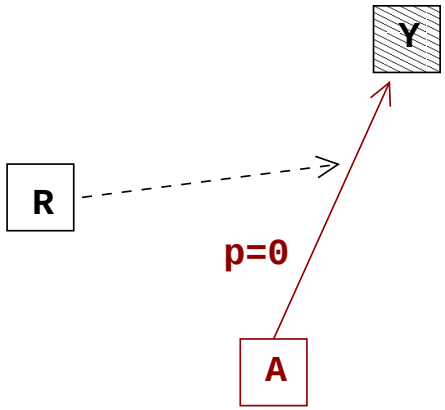
$$P(Y, B) = P(Y|B)P(B)$$

DBNs - switching parents

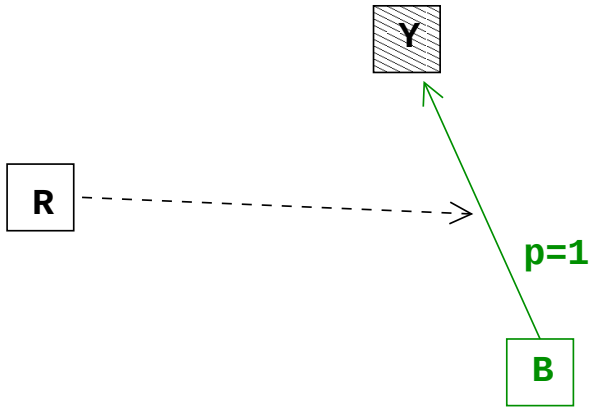


$$P(Y, A, B) = P(R = 0)P(Y|A)P(A) + P(R = 1)P(Y|B)P(B)$$

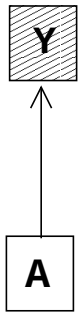
DBNs - switching parents



DBNs - switching parents

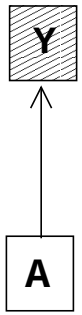


DBNs - Observed variables



Joint distribution here is $P(Y, A) = P(Y|A)P(A)$.

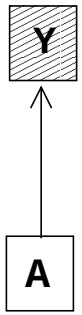
DBNs - Observed variables



Joint distribution here is $P(Y, A) = P(Y|A)P(A)$.

- Y is discrete and observed

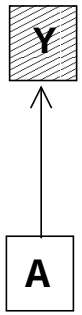
DBNs - Observed variables



Joint distribution here is $P(Y, A) = P(Y|A)P(A)$.

- Y is discrete and observed
- $P(Y|A)$ is encoded as a conditional probability table (CPT)

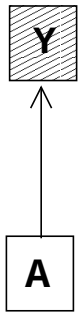
DBNs - Observed variables



Joint distribution here is $P(Y, A) = P(Y|A)P(A)$.

- Y is discrete and observed
- $P(Y|A)$ is encoded as a conditional probability table (CPT)
  - gives a value for $P(Y = y|A = a)$ for $y = 1, \ldots, N_Y, a = 1, \ldots, N_A$

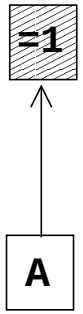
DBNs - Observed variables



Joint distribution here is $P(Y, A) = P(Y|A)P(A)$.

- Y is discrete and observed
- $P(Y|A)$ is encoded as a conditional probability table (CPT)
  - gives a value for $P(Y = y|A = a)$ for $y = 1, \ldots, N_Y, a = 1, \ldots, N_A$
  - given $A = a$, we have a probability mass function for Y, i.e. $\Sigma_{y=1,\ldots,N_Y} P(Y = y|A = a) = 1$

DBNs - virtual evidence (VE)



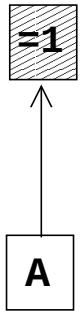
Joint distribution here is $P(Y, A) = f_Y(A)P(A)$.

DBNs - virtual evidence (VE)

Joint distribution here is $P(Y, A) = f_Y(A)P(A)$.

- In spirit of hybrid ANN/HMM

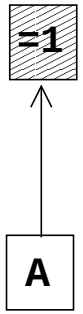
DBNs - virtual evidence (VE)



Joint distribution here is $P(Y, A) = f_Y(A)P(A)$.

- In spirit of hybrid ANN/HMM
- Rather than observing Y, virtual evidence (VE) $f_Y(A = a), a = 1, \ldots, N_A$ is read in instead.

DBNs - virtual evidence (VE)



Joint distribution here is $P(Y, A) = f_Y(A)P(A)$.

- In spirit of hybrid ANN/HMM
- Rather than observing Y, virtual evidence (VE) $f_Y(A = a), a = 1, \ldots, N_A$ is read in instead.
- VE can be from any model which can assign likelihood/posterior to each of the discrete values of A.

DA decoding DBN - elements

The main elements of the DA decoder are:

- Random variables (RV)

DA decoding DBN - elements

The main elements of the DA decoder are:

- Random variables (RV)
  - DA-internal word: binary RV, corresponds to whether current word is final or internal to a DA

DA decoding DBN - elements

The main elements of the DA decoder are:

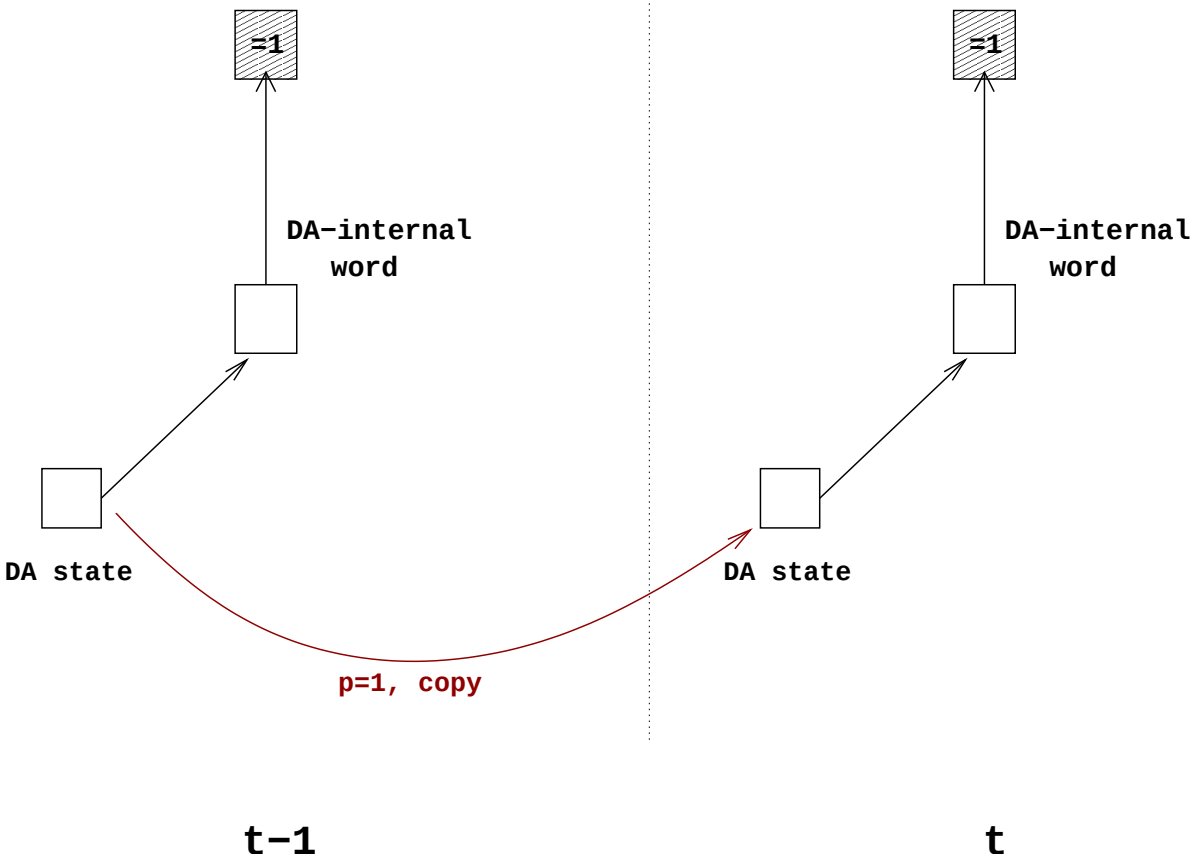
- Random variables (RV)
  - DA-internal word: binary RV, corresponds to whether current word is final or internal to a DA
  - DA state: 5 levels, one for each DA class

DA decoding DBN - elements

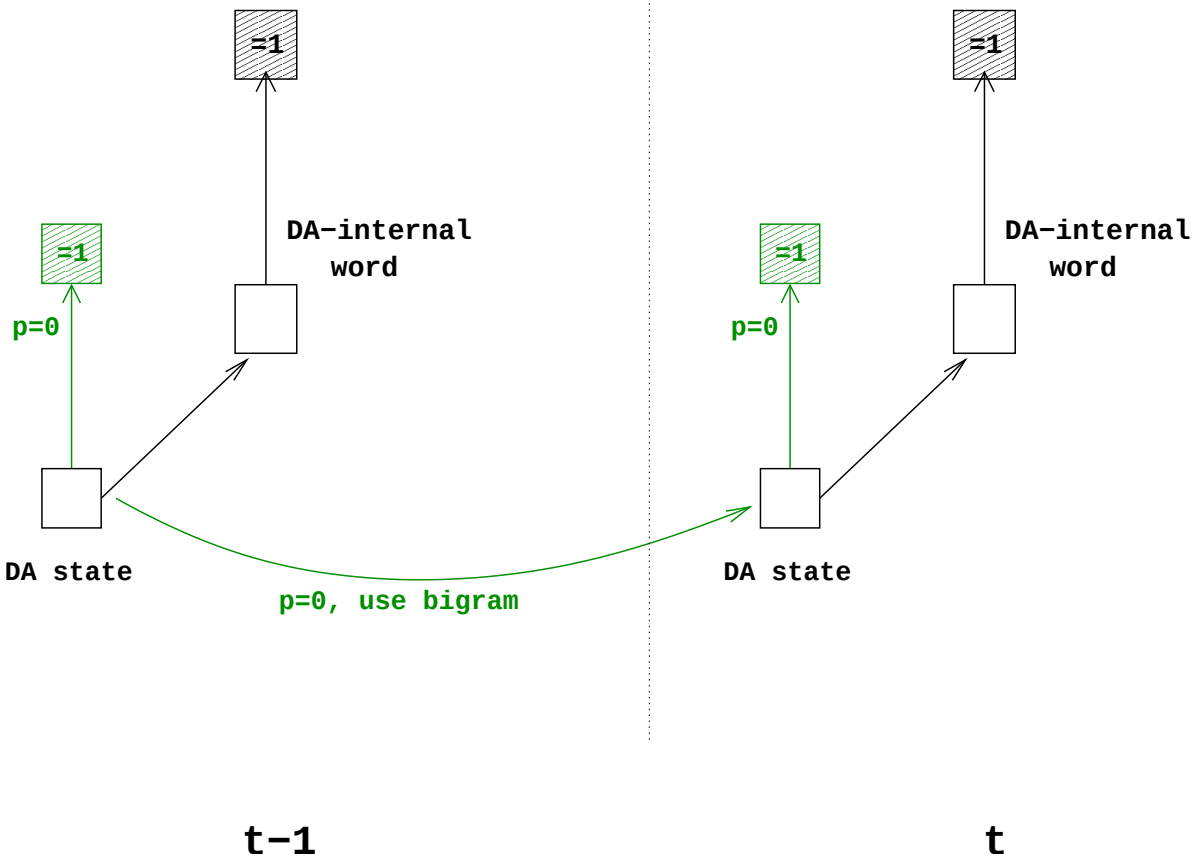
The main elements of the DA decoder are:

- Random variables (RV)
  - DA-internal word: binary RV, corresponds to whether current word is final or internal to a DA
  - DA state: 5 levels, one for each DA class
- Virtual evidence (VE) is available for each of these.

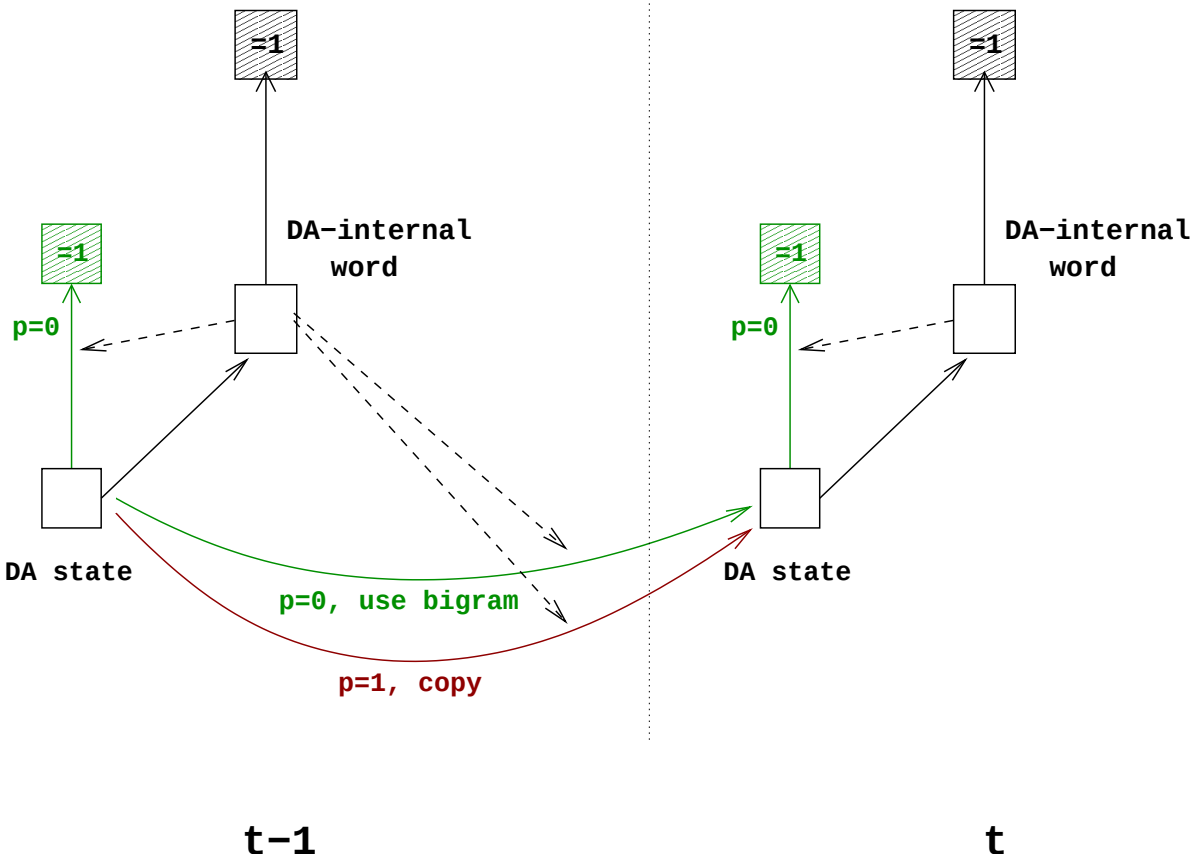
DBN case 1: no DA transition, so copy state



DBN case 2: DA transition, so use DA bigram



DBN: full model with switching parents



DBN - results

model	strict	F-measure
CRF, threshold	61.6	52.5
DBN	61.1	52.7

Data - with activity information

transcript	channel 1 state	channel 1 VE P(final)	channel 1 VE P(internal)	channel 2 state	channel 2 VE P(final)	channel 2 VE P(internal)
did	-	-	-	q	0.01	0.99
you	-	-	-	q	0.02	0.98
take	-	-	-	q	0.01	0.99
a	-	-	-	q	0.01	0.99
day	-	-	-	q	0.08	0.92
off?	-	-	-	**Q**	**0.95**	**0.05**
yeah	s	0.01	0.99	-	-	-
that's	s	0.02	0.98	-	-	-
right.	**S**	**0.98**	**0.02**	-	-	-

DBN with activity information

- Previous model included:

DBN with activity information

- Previous model included:
  - DA-internal word + associated VE

DBN with activity information

- Previous model included:
  - DA-internal word + associated VE
  - DA state + associate VE

DBN with activity information

- Previous model included:
  - DA-internal word + associated VE
  - DA state + associate VE
- Now add:

DBN with activity information

- Previous model included:
  - DA-internal word + associated VE
  - DA state + associate VE
- Now add:
  - DA state active - binary variable to say whether the state is inactive (speaker quiet) or active (has an associated word)

DBN with activity information

- Previous model included:
  - DA-internal word + associated VE
  - DA state + associate VE
- Now add:
  - DA state active - binary variable to say whether the state is inactive (speaker quiet) or active (has an associated word)
  - DA state memory - holds the last DA state prior to an inactive region

DBN with activity information - state behaviours

Now we have inactive regions, so there are some extra possibilities for the state behaviour:

- frame inactive - nil level

DBN with activity information - state behaviours

Now we have inactive regions, so there are some extra possibilities for the state behaviour:

- frame inactive - nil level
- no transition - stay in the same state

DBN with activity information - state behaviours

Now we have inactive regions, so there are some extra possibilities for the state behaviour:

- frame inactive - nil level
- no transition - stay in the same state
- first frame after an inactive period, though memory alive - copy from memory

DBN with activity information - state behaviours

Now we have inactive regions, so there are some extra possibilities for the state behaviour:

- frame inactive - nil level
- no transition - stay in the same state
- first frame after an inactive period, though memory alive - copy from memory
- first frame after an inactive period, no memory, so use unigram

DBN with activity information - state behaviours

Now we have inactive regions, so there are some extra possibilities for the state behaviour:

- frame inactive - nil level
- no transition - stay in the same state
- first frame after an inactive period, though memory alive - copy from memory
- first frame after an inactive period, no memory, so use unigram
- transition within active period, use bigram

DBN with activity information - results

model	strict	F-measure
CRF, threshold	61.6	52.5
DBN	61.1	52.7
DBN + activity information	61.1	52.7

DBN with activity information - results

model	strict	F-measure
CRF, threshold	61.6	52.5
DBN	61.1	52.7
DBN + activity information	61.1	52.7

- No improvement in either measure from using activity information.

DBN with activity information - results

model	strict	F-measure
CRF, threshold	61.6	52.5
DBN	61.1	52.7
DBN + activity information	61.1	52.7

- No improvement in either measure from using activity information.
  - CRF uses duration of the pause between two words from the same speaker as a feature, so the activity information is already implicit.

DBN with activity information - results

model	strict	F-measure
CRF, threshold	61.6	52.5
DBN	61.1	52.7
DBN + activity information	61.1	52.7

- No improvement in either measure from using activity information.
  - CRF uses duration of the pause between two words from the same speaker as a feature, so the activity information is already implicit.
  - this model serves as useful sanity check for the joint multichannel version.

DBNs - multichannel decoding

All the RVs are encoded in CPP macros, so to decode/train the model, a DBN is instantiated according to the number of channels (speakers) in the meeting.

DBNs - multichannel decoding

All the RVs are encoded in CPP macros, so to decode/train the model, a DBN is instantiated according to the number of channels (speakers) in the meeting.

- Create DBN with activity information for each channel.

DBNs - multichannel decoding

All the RVs are encoded in CPP macros, so to decode/train the model, a DBN is instantiated according to the number of channels (speakers) in the meeting.

- Create DBN with activity information for each channel.
- Add a 'system state'

DBNs - multichannel decoding

All the RVs are encoded in CPP macros, so to decode/train the model, a DBN is instantiated according to the number of channels (speakers) in the meeting.

- Create DBN with activity information for each channel.
- Add a 'system state'
  - unique state for each combination of DA states which are present at a given time

DBNs - multichannel decoding

All the RVs are encoded in CPP macros, so to decode/train the model, a DBN is instantiated according to the number of channels (speakers) in the meeting.

- Create DBN with activity information for each channel.
- Add a 'system state'
  - unique state for each combination of DA states which are present at a given time
  - simply modelled by a discrete probability distribution

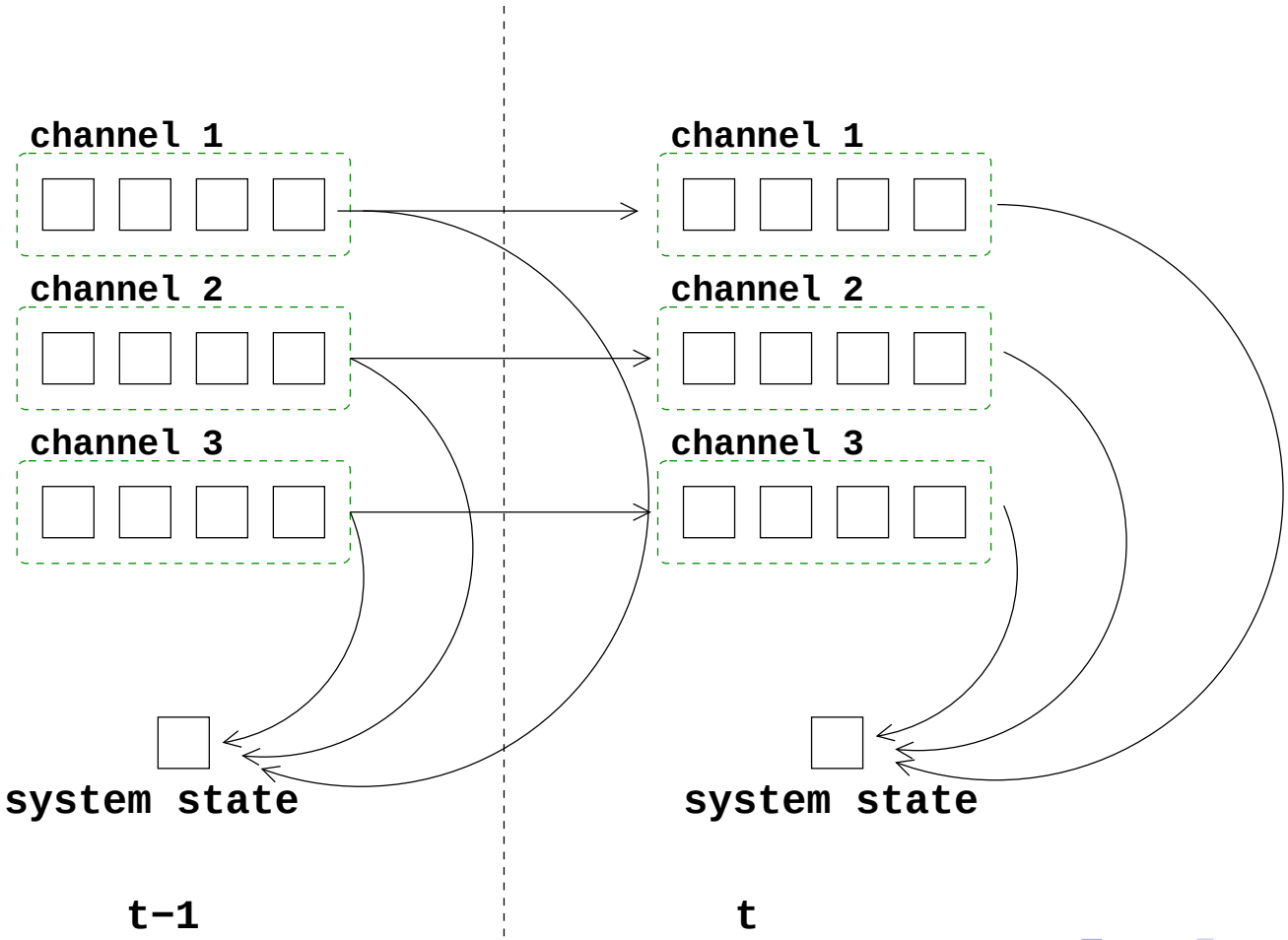
DBNs - multichannel decoding

- State transitions are modelled as previously.

DBNs - multichannel decoding

- State transitions are modelled as previously.
- All CPTs are treated as shared parameters, rather than being specific to a single channel.

DBNs - multichannel decoding, 3 channel example



DBNs - multichannel decoding - results

model	strict	F-measure
CRF, threshold	61.6	52.5
single-channel DBN	61.1	52.7
multi-channel DBN	61.1	52.6

DBNs - multichannel decoding - analysis

channels	strict error		f-measure	
	single	multi	single	multi
3	68.7	**66.3**	47.0	**49.2**
4	61.3	**60.8**	53.3	**53.7**
5	63.3	**62.1**	50.3	**50.6**
6	58.0	**57.2**	**56.0**	55.9
7	**63.1**	63.9	**51.1**	50.7
8	**56.7**	57.5	**54.6**	54.0

- Joint multi-channel decoding gives improved results where there are fewer channels.

DBNs - multichannel decoding - analysis

channels	strict error		f-measure	
	single	multi	single	multi
3	68.7	**66.3**	47.0	**49.2**
4	61.3	**60.8**	53.3	**53.7**
5	63.3	**62.1**	50.3	**50.6**
6	58.0	**57.2**	**56.0**	55.9
7	**63.1**	63.9	**51.1**	50.7
8	**56.7**	57.5	**54.6**	54.0

- Joint multi-channel decoding gives improved results where there are fewer channels.
- Possibly due to pruning errors during decoding

DBNs - multichannel decoding - analysis

channels	strict error		f-measure	
	single	multi	single	multi
3	68.7	**66.3**	47.0	**49.2**
4	61.3	**60.8**	53.3	**53.7**
5	63.3	**62.1**	50.3	**50.6**
6	58.0	**57.2**	**56.0**	55.9
7	**63.1**	63.9	**51.1**	50.7
8	**56.7**	57.5	**54.6**	54.0

- Joint multi-channel decoding gives improved results where there are fewer channels.
- Possibly due to pruning errors during decoding
  - 6, 7, and 8 -factorial models lead to very large state-spaces

DBNs - multichannel decoding - analysis

channels	strict error		f-measure	
	single	multi	single	multi
3	68.7	**66.3**	47.0	**49.2**
4	61.3	**60.8**	53.3	**53.7**
5	63.3	**62.1**	50.3	**50.6**
6	58.0	**57.2**	**56.0**	55.9
7	**63.1**	63.9	**51.1**	50.7
8	**56.7**	57.5	**54.6**	54.0

- Joint multi-channel decoding gives improved results where there are fewer channels.
- Possibly due to pruning errors during decoding
  - 6, 7, and 8 -factorial models lead to very large state-spaces
  - need to find better triangulation

DBNs - multichannel decoding - future

- Many possible modifications in model to examine, e.g. conditioning speaker state transitions on the system state

DBNs - multichannel decoding - future

- Many possible modifications in model to examine, e.g. conditioning speaker state transitions on the system state
- CPTs (e.g. state transitions) are treated as system-wide, though can be posed as mixture distributions

DBNs - multichannel decoding - future

- Many possible modifications in model to examine, e.g. conditioning speaker state transitions on the system state
- CPTs (e.g. state transitions) are treated as system-wide, though can be posed as mixture distributions
  - investigate using these to infer a speaker's characteristics, e.g. role in the meeting.

DBNs - multichannel decoding - future

- Many possible modifications in model to examine, e.g. conditioning speaker state transitions on the system state
- CPTs (e.g. state transitions) are treated as system-wide, though can be posed as mixture distributions
  - investigate using these to infer a speaker's characteristics, e.g. role in the meeting.
- Train a set of CRFs to classify the start of dialogue acts

DBNs - multichannel decoding - future

- Many possible modifications in model to examine, e.g. conditioning speaker state transitions on the system state
- CPTs (e.g. state transitions) are treated as system-wide, though can be posed as mixture distributions
  - investigate using these to infer a speaker's characteristics, e.g. role in the meeting.
- Train a set of CRFs to classify the start of dialogue acts
  - some classes (e.g. what/why questions) are better formed at the start than the end of the DA.

The End

Thanks!